COMSOL Multiphysics®

Reference Manual

VERSION 4.4

COMSOL

COMSOL Multiphysics Reference Manual

Version: November 2013 COMSOL 4.4

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries, contact Technical Support, or search for an address and phone number. You can also visit the Worldwide Sales Offices page at www.comsol.com/contact/offices for address and contact information.

If you need to contact Support, an online request form is located at the COMSOL Access page at www.comsol.com/support/case.

Other useful links include:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/support/download
- Product Updates: www.comsol.com/support/updates
- COMSOL Community: www.comsol.com/community
- Events: www.comsol.com/events
- COMSOL Video Center: www.comsol.com/video
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part number: CM020005

Contents

Chapter 1: Introduction

Chapter 2: The COMSOL Modeling Environment

Chapter 3: Building a COMSOL Model

Chapter 4: Customizing the COMSOL Desktop

Chapter 5: Definitions

Functions 239

Component Couplings 257

Coordinate Systems 273

Identity and Contact Pairs 283

Probes 288

Chapter 6: Visualization and Selection Tools

Chapter 7: Geometry Modeling and CAD Tools

Geometry Operations **397**

Virtual Geometry and Mesh Control Operations **431**

Geometry Modeling Examples **438**

Chapter 8: Meshing

Chapter 9: Materials

Chapter 10: The AC/DC Interfaces

Chapter 11: Pressure Acoustics

Chapter 12: Chemical Species Transport

Chapter 13: Fluid Flow

Chapter 14: Heat Transfer Modeling

Chapter 15: Structural Mechanics

Chapter 16: Equation-Based Modeling

The PDE Interfaces **779**

Theory for the Wave Form PDE **799**

The Wave Form PDE Interface **803**

About Auxiliary Equation-Based Nodes **809**

Chapter 17: Sensitivity Analysis

Chapter 18: Deformed Geometry and Moving Mesh

Chapter 19: Studies and Solvers

Computing a Solution 928

Solution Operation Nodes and Solvers 938

Solution Attribute Nodes 969

Chapter 20: Results Analysis and Plots

Derived Values and Tables **1130**

Exporting Data and Images **1140**

Reports **1150**

Chapter 21: Running COMSOL

Chapter 22: Glossary

1

Introduction

Welcome to COMSOL Multiphysics®! This book details features and techniques that help you throughout all of your COMSOL modeling in Version 4.4 using the COMSOL Desktop® environment. For example, detailed information about how to build model geometries in COMSOL, how to create a mesh for the finite elements, how to create parameters and variables used within a model, how to add the physics and material properties, and how to solve and display the results, are all explained. The full set of documentation shows you, step by step, how to tap into many functions and capabilities available in the COMSOL environment.

This introductory chapter provides an overview of COMSOL Multiphysics and its product family, documentation set, and other resources.

Version 4.4 brings all-new COMSOL desktop environments for Windows® and as a cross-platform version for Linux® and Mac with user interface components such as ribbon toolbars in Windows for easy, single-click access to most functionality and new, easier-to-use ways to select geometric entities and add, for example, boundary conditions, materials, and physics to the picked entities. See About This Release of COMSOL Multiphysics for information about the changes. This release also includes an extended and improved *Physics Builder* for creating your own specialized physics interfaces.

About COMSOL Multiphysics

COMSOL Multiphysics is a powerful interactive environment used to model and solve all kinds of scientific and engineering problems. The software provides a powerful integrated desktop environment with a *Model Builder* that gives you a full overview of the model and access to all functionality. With COMSOL Multiphysics you can easily extend conventional models for one type of physics into multiphysics models that solve coupled physics phenomena—and do so simultaneously. Accessing this power does not require an in-depth knowledge of mathematics or numerical analysis.

Using the built-in *physics interfaces* and the advanced support for material properties, you can build models by defining the relevant physical quantities—such as material properties, loads, constraints, sources, and fluxes—rather than by defining the underlying equations. You can always apply these variables, expressions, or numbers directly to solid and fluid domains, boundaries, edges, and points independently of the computational mesh. COMSOL then internally compiles a set of equations representing the entire model.

You access the power of COMSOL as a standalone product through a flexible graphical user interface (GUI) or by script programming in Java® or the MATLAB® language (this requires a LiveLink™ *for* MATLAB® license).

Using these physics interfaces, you can perform various types of studies including:

- Stationary and time-dependent (transient) studies
- Linear and nonlinear studies
- Eigenfrequency, modal, and frequency response studies

When solving the models, COMSOL assembles and solves the problem using a set of advanced numerical analysis tools. The software runs the analysis together with adaptive meshing (if selected) and error control using a variety of numerical solvers. The studies can make use of multiprocessor systems and cluster computing, and you can run batch jobs and parametric sweeps.

COMSOL creates *sequences* to record all steps that create the geometry, mesh, studies and solver settings, and visualization and results presentation. This makes it easy to parameterize any part of the model; simply change a node in the model tree and re-run the sequences. The program remembers and reapplies all other information and data in the model.

REAL-WORLD APPLICATIONS

Partial differential equations (PDEs) form the basis for the laws of science and provide the foundation for modeling a wide range of scientific and engineering phenomena. You can use COMSOL in many application areas, including:

- Acoustics
- Bioscience
- Chemical reactions
- Corrosion and corrosion protection
- Diffusion
- Electrochemistry
- Electromagnetics
- Fatigue analysis
- Fluid dynamics
- Fuel cells and electrochemistry
- Geophysics and geomechanics
- Heat transfer
- Microelectromechanical systems (MEMS)
- Microfluidics
- Microwave engineering
- Multibody dynamics
- Optics
- Particle tracing
- Photonics
- Plasma physics
- Porous media flow
- Quantum mechanics
- Radio-frequency components
- Semiconductor devices
- Structural mechanics
- Transport phenomena
- Wave propagation

Many real-world applications involve simultaneous couplings in a system of PDEs—*multiphysics*. For instance, the electric resistance of a conductor often varies with temperature, and a model of a conductor carrying current should include resistive-heating effects. The Multiphysics Modeling Approaches section discusses multiphysics modeling techniques. Many predefined physics interfaces provide easy-to-use entry points for common multiphysics applications.

In its base configuration, COMSOL offers modeling and analysis power for many application areas. For several of the key application areas there are also optional modules. These application-specific modules use terminology and solution methods specific to the particular discipline, which simplifies creating and analyzing models. The modules also include comprehensive model libraries with example models that show the use of the product within its application areas.

About This Release of COMSOL Multiphysics

For use on the Windows® platform, the COMSOL 4.4 release introduces a ribbon layout, a style familiar to Microsoft® Office users and integrated into many other software designs. The ribbon-style layout is intuitive and makes it easy to locate similar and frequently used features. For the Linux® and Mac platforms, COMSOL 4.4 includes extended toolbars that provide almost identical single-click access to most functionality in the software.

For experienced COMSOL users, the basic functionality has not changed but access to some menus and physics features have been moved. Many changes are related to different menu paths and icons. The COMSOL Desktop has the same flexibility but in addition to the ribbon toolbars, the fonts and icons have changed.

!	The ribbon layout is only available with Windows operating systems. For Mac and Linux users there is a similar approach to how the toolbars are displayed and used, including new icons and contextual menus, but it does not look the same, nor it referred to as a ribbon.

For all users, the new layout highlights the most used functionality; menu items are grouped with similar tasks and actions. The features are still also available when you right-click a node in the model tree to open the context menu—there are often several ways to complete the same task. As before, the available options in any context menu or ribbon toolbar are based on where you are in the model, at what stage of the process of solving, or whether it is relevant to the model being built. In other words, it is contextual.

Also see the *COMSOL Multiphysics Release Notes* for more information about this release.

DIFFERENT INSTRUCTIONS FOR DIFFERENT OPERATING SYSTEMS

The addition of the ribbon for Windows users means that there are slightly different instructions about how to access some features compared to Mac or Linux users. When specific instructions are included about where to find a particular feature, the instructions distinguish between the operating systems using different icons.

- Where there are no differences, the icons are not used.
- Where there are minor differences in appearance or accessibility, but the functionality is the same, no icons are used.
- In general, instructions for all platform imply that the feature is available from a named toolbar. For example, the **Physics** toolbar, **Mesh** toolbar, or **Geometry** toolbar. See Toolbars and Keyboard Shortcuts for information about each toolbar.

Win

A *ribbon tab*, *ribbon group*, or *modal ribbon tab*, are available in the Windows version. See Figure 2-1 for an example of the Windows **Home** ribbon.

Mac

Linux

The *Model Toolbar* and *Contextual Toolbar* are available in the cross-platform version, primarily for Mac and Linux users. See Figure 2-9 for an example of these toolbars.

ABOUT THE SCREEN SHOTS USED IN THIS MANUAL

The screen shots used throughout this reference manual are captured using the Windows platform except where there are clear differences other than fonts or cosmetic appearance.

NEW APPROACH TO MULTIPHYSICS MODELING

In previous versions of COMSOL Multiphysics, specific multiphysics interfaces, such as The Joule Heating Interface, were added to the Model Builder. Now, a predefined multiphysics coupling approach is used, improving the flexibility and design options for your modeling. However, this is not yet available for all multiphysics interfaces for all modules. In the future all multiphysics interfaces will use this approach but for this release, only interfaces with a Heat Transfer interface have changed. For specific details, see Multiphysics Modeling Approaches.

The COMSOL Modules and Interfacing Options

The optional modules, including interfacing options such as the CAD Import Module and bidirectional interfaces such as the LiveLink™ products, are optimized for specific application areas and offer discipline-standard terminology and physics interfaces. For some modules, additional material libraries, specialized solvers, element types, and visualization tools are also available.

💡	For up-to-date module availability, product descriptions, and a specification chart, go to www.comsol.com/products.

COMSOL Documentation and Help

About the Documentation Set

The full documentation set that ships with COMSOL Multiphysics consists of the following titles:

- *Introduction to COMSOL Multiphysics*—information about version 4.4 and how to build models using the desktop environment, including quick references to keyboard shortcuts and common commands and functions.
- *COMSOL License Agreement.*
- *COMSOL Installation Guide*—besides covering various installation options, it describes system requirements and how to configure and run the COMSOL software on different platforms, including client/server architectures as well as shared-memory and distributed (cluster) parallel versions.
- *COMSOL Multiphysics Reference Manual*—this book, which covers the functionality of COMSOL Multiphysics across its entire range from geometry modeling to results evaluation and visualization, including the physics interfaces for physics and equation-based modeling. It serves as a tutorial and a reference guide to use COMSOL Multiphysics. This book reviews geometry, mesh, solver, and results functionality and provides detailed information about the settings and options. Additionally, it describes some advanced functionality and settings in COMSOL Multiphysics and provides background material and references.
- *COMSOL API for use with Java® Reference Manual*—this book provides details about features and techniques that help you control COMSOL Multiphysics using its application programming interface (API). The COMSOL API can be used from a standalone Java application as well as from MATLAB® using the LiveLink™ *for* MATLAB® interface.
- The *Physics Builder Manual* provides documentation related to the Physics Builder.

In addition, each of the optional modules includes a manual as described in The COMSOL Modules and Interfacing Options. The documentation for the optional CAD Import Module and LiveLinks to CAD packages is available in separate manuals, and the documentation for the optional Material Library in the *Material Library User's Guide*.

The *COMSOL LiveLink™ for MATLAB® User's Guide* shows how to access the capabilities of COMSOL from the MATLAB programming environment.

A number of Internet resources provide more information about COMSOL, including licensing and technical information. The electronic documentation, topic-based (or context-based) help, and the Model Libraries are all accessed through the COMSOL Desktop.

!	If you are reading the documentation as a PDF file on your computer, the blue links do not work to open a model or content referenced in a different guide. However, if you are using the Help system in COMSOL Multiphysics, these links work to other modules (as long as you have a license), model examples, and documentation sets.

CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at info@comsol.com.

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and case number is sent to you by email.

COMSOL WEBSITES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
Support Center	www.comsol.com/support
Product Download	www.comsol.com/support/download
Product Updates	www.comsol.com/support/updates
COMSOL Community	www.comsol.com/community
Events	www.comsol.com/events
COMSOL Video Gallery	www.comsol.com/video
Support Knowledge Base	www.comsol.com/support/knowledgebase

The Help Window and Topic-Based Help

The **Help** window is useful as it is connected to many of the features on the COMSOL Desktop. This concept is called *topic-based help* or *context help*. You can also search all the HTLM documentation content from this window.

!

The **Help** system automatically starts a web server using port 8090 on the computer where COMSOL is installed. Depending on the security settings, you might get a question to allow that port to be used the first time the help system is started.

The operating system might also issue a firewall security warning. To use Help, allow COMSOL access through the firewall.

To learn more about a node in the Model Builder, or a window on the Desktop, click to highlight a node or window, then press F1. The Help window opens and displays the topic information about the selected feature.

OPENING THE HELP WINDOW AND THE TOPIC-BASED HELP

To open the **Help** window:

- Press F1.
- On the main toolbar, click **Help** (?).
- From the main menu, select **File>Help** (Windows) **Help>Help** (Linux and Mac)
- Right-click any node in the **Model Builder** and select **Help**.

ABOUT USING THE F1 KEY TO ACCESS CONTEXT HELP

To display topic-based (context) information in the **Help** window, on the COMSOL Desktop:

- Click to highlight a node in the Model Builder tree. For example, the **Component** or **Geometry** node.

- Click a window tab, for example, **Model Builder**, **Add Study**, or **Messages**.
- For Windows users, hover over toolbar buttons to display a *tooltip*. At the same time you can press F1 to display more detail. This is only applicable to buttons, not submenus.

> In some cases you need to refocus the context help on its target before pressing F1. Try clicking to highlight a node, a window, or the button or click to focus on the Help window, hover over a toolbar button (Windows only) and press F1.

TABLE 1-1: THE HELP AND DOCUMENTATION TOOLBARS

BUTTON	NAME	DESCRIPTION
	Home	Returns to the COMSOL Documentation window home page. Only available for The Documentation Window.
	Contents	Open a tree-based menu of the COMSOL documentation. It is the HTML version of the documentation that is also accessed from the Documentation window. Only available for the Help window. When you select a node in the table of contents tree the corresponding topic is shown on the Topic page.
🔍	Search	On the Help window, click to open the search engine to look for contents in the COMSOL documentation. Search results are shown sorted by product. On the Documentation window, enter Search terms in the field and choose the Search scope—All documents, Selected only, or Model library. See Searching Help and Documentation Content for more information about search terms you can use.
	Topic	On the Help window, jump directly to the information about a node or window that you have clicked in the COMSOL Desktop. This is an interactive environment. Click a node to update the contents instantly.
🔒	Sticky Help	On the Help window, click the Sticky Help button to lock the current help window (the icon is highlighted 🔒), which can be useful to keep some help topic or model instruction active, or to release the window and view topic-based (context) help when a node or window is clicked.
←	Back	Navigates backward to the topics previously selected. For the Help window, move back or forward in the browser history on the Search or Topic pages.
→	Forward	Navigates forward within a topic, but only to the end of the current list.
>	Next	Navigates forward to the next topic in the order displayed.
<	Previous	Navigates backward to the previous contents topic.

CHANGING THE DEFAULT HELP SETTINGS

To edit the following settings, open The Preferences Dialog Box and click **General**.

Setting the Help Mode

From the **Help mode** list, select **Integrated** (the default) to show the help contents in the **Help** window that is integrated in the COMSOL Desktop environment, or select **Web browser** to display the help contents in a separate browser. For the integrated mode, the **Show PDF-files in new browser window** check box controls how PDF documents are launched. Click to clear this check box to use your default system browser's settings. Edit the **Documentation root directory** file path as required.

The default file paths are based on the platform:

- On Windows `C:\Program Files\COMSOL\COMSOL44\doc`, or generically `COMSOL44\doc`.
- For Mac and Linux, under the main COMSOL installation directory: `COMSOL44/doc`.

Selecting a Web Browser

Under **Web browser** (Windows and Linux only), you can choose which browser COMSOL should use to show pages on the COMSOL website and help contents when using the web browser help mode (see below). The following settings are available:

- On Windows: Choose the **Program** setting **System default** to use the default system web browser. Alternatively, choose **Custom** and then give the path to an **Executable** location for a different browser installed on your computer.
- On Linux: Type the path to the web browser directly in the **Executable** field, or click the **Browse** button and then point to the executable file on the file system.

On Mac, this setting is not available and COMSOL always uses the system's default web browser.

The Documentation Window

Win

To open the **Documentation** window:

- Press Ctrl+F1.
- From the **File** menu, select **Help>Documentation**.

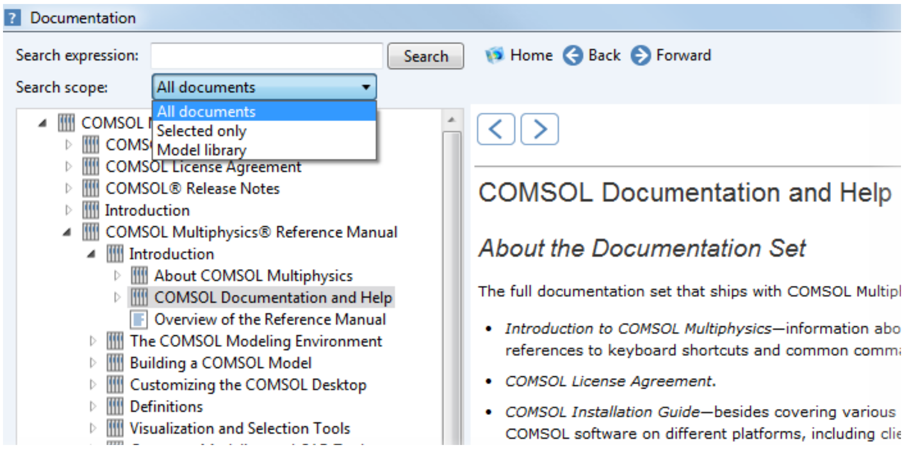
Mac

Linux

To open the **Documentation** window:

- Press Ctrl+F1
- On the main toolbar, click the **Documentation** (📘) button.
- From the **Help** menu, select **Documentation**.

In the Documentation window, you can navigate to PDF or HTML versions of the documentation (availability is based on your license), as well as search all the documentation, save or open PDFs, or view the HTML content in this window. There are different ways to access the same information using either the left-hand side (Figure 1-1) or right-hand side (Figure 1-2) of the window.



Documentation

Search expression: Search | Home | Back | Forward

Search scope: All documents

- All documents
- Selected only
- Model library

- COMSOL M
  - COMS
  - COMSOL License Agreement
  - COMSOL® Release Notes
  - Introduction
  - COMSOL Multiphysics® Reference Manual
    - Introduction
      - About COMSOL Multiphysics
      - COMSOL Documentation and Help
      - Overview of the Reference Manual
    - The COMSOL Modeling Environment
    - Building a COMSOL Model
    - Customizing the COMSOL Desktop
    - Definitions
    - Visualization and Selection Tools

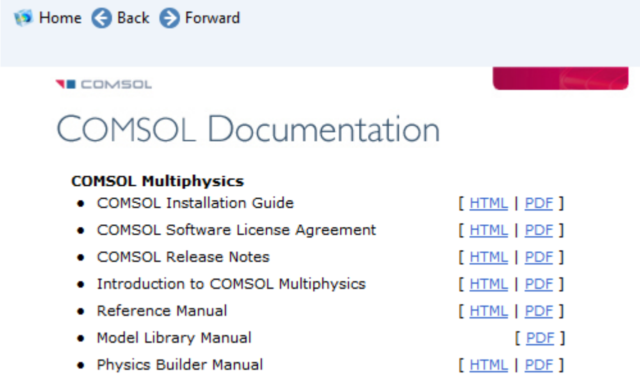
COMSOL Documentation and Help

About the Documentation Set

The full documentation set that ships with COMSOL Multipl

- *Introduction to COMSOL Multiphysics*—information abo references to keyboard shortcuts and common comm
- *COMSOL License Agreement.*
- *COMSOL Installation Guide*—besides covering various COMSOL software on different platforms, including clie

Figure 1-1: The left side of the Documentation window. Based on your license, links to the HTML versions of the product documentation are accessed and can be browsed in the tree. When you click a topic in the tree, the information displays to the right. You can also adjust the search scope.



Home | Back | Forward

COMSOL

COMSOL Documentation

COMSOL Multiphysics

COMSOL Installation Guide	[HTML | PDF]
COMSOL Software License Agreement	[HTML | PDF]
COMSOL Release Notes	[HTML | PDF]
Introduction to COMSOL Multiphysics	[HTML | PDF]
Reference Manual	[HTML | PDF]
Model Library Manual	[PDF]
Physics Builder Manual	[HTML | PDF]

Figure 1-2: The right side of the Documentation window. Based on your license, links to PDF and HTML versions of the product documentation are accessed from this window. When you click HTML, it jumps to the first page of the documentation for that product. When you click PDF, you have the option to Open or Save a full PDF version of that document.

- The Help Window and Topic-Based Help
- Table 1-1 for a list of the Documentation toolbar buttons.

Searching Help and Documentation Content

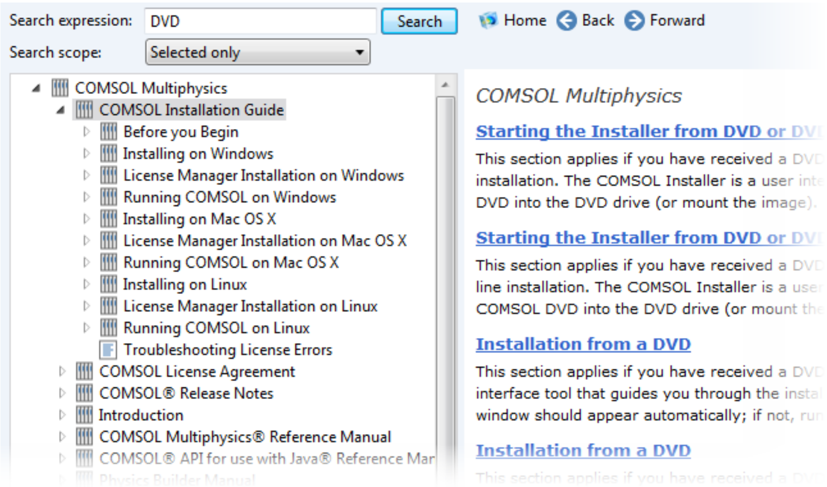
After you open The Help Window and Topic-Based Help, click the Search button ( ) to open the search engine to and search the HTML content. Search results are shown sorted by product.

You can also search The Documentation Window. where you enter search terms in the field and choose a Search scope—All documents, Selected only, or Model library.

The first search can take a couple of minutes while the search index is generated.

SEARCHING THE DOCUMENTATION

On the **Documentation** window, you can adjust the **Search scope** (see Figure 1-1). Enter a search term and then select **All documents**, **Selected only**, or **Model library** from the list to narrow or expand the search scope as required. For **Selected only** (Figure 1-3), first click a branch in the tree (for example, **COMSOL Installation Guide**) and then the search includes all the documents below the selected node until the beginning of the next branch, in this example it searches until the end of the Troubleshooting License Errors section.



Search expression: DVD

Search scope: Selected only

COMSOL Multiphysics
COMSOL Installation Guide
Before you Begin
Installing on Windows
License Manager Installation on Windows
Running COMSOL on Windows
Installing on Mac OS X
License Manager Installation on Mac OS X
Running COMSOL on Mac OS X
Installing on Linux
License Manager Installation on Linux
Running COMSOL on Linux
Troubleshooting License Errors
COMSOL License Agreement
COMSOL® Release Notes
Introduction
COMSOL Multiphysics® Reference Manual
COMSOL® API for use with Java® Reference Mar
Physics Builder Manual

COMSOL Multiphysics

Starting the Installer from DVD or DV

This section applies if you have received a DVD installation. The COMSOL Installer is a user inte DVD into the DVD drive (or mount the image).

Starting the Installer from DVD or DV

This section applies if you have received a DVD line installation. The COMSOL Installer is a use COMSOL DVD into the DVD drive (or mount the

Installation from a DVD

This section applies if you have received a DVD interface tool that guides you through the insta window should appear automatically; if not, ru

Installation from a DVD

This section applies if you have received a DVD

Figure 1-3: When searching in the Documentation window, choose a Search scope to search only a selected portion of the documentation, all the documentation, or only the model libraries.

SEARCH PARAMETERS FOR HELP AND THE DOCUMENTATION

Some examples of search parameters you can use:

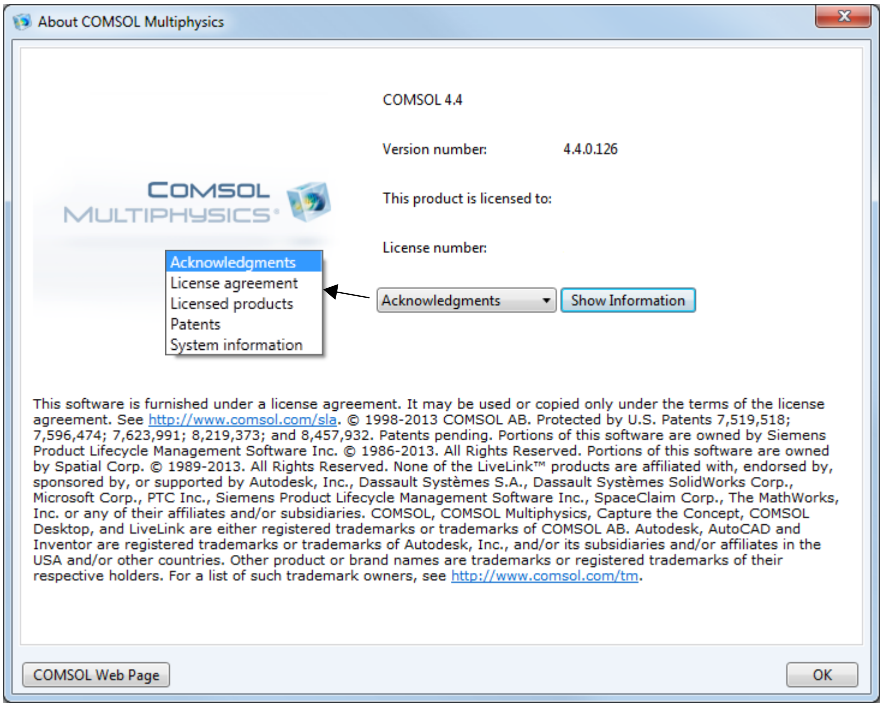
TABLE 1-2: SEARCH PARAMETERS FOR THE COMSOL HELP SYSTEM

OPERATOR	EXAMPLE	SEARCH RESULT EXAMPLE
&&, AND	block && cone block AND cone	Results include all instances of the words.
OR, ||	block OR cone block || cone	Results include any of the listed words.
+, –	+block –cone	Search for one term (+block) but not the other term (–cone).
“, ~	“plot line”~10	Search for the words enclosed in the quotation marks (plot line) within (~) a certain number of words (10) from each other.
~	ecentric~	Search for something “almost” spelled in a particular way. For example, ecentric. The results include eccentric cone.
?	h?t	Use in a search query to mean exactly one character. For example, search for all instances of hat, hit, or hut where ? represents a, i, or u, or any other letter between h and t.
*	strain* strain*d	Search for any word that starts with “strain”. Results include strain-based, strain-rate, or strain, for example. The asterisk represents any number of characters. If the asterisk is used in the middle of the word, it searches for one letter between “n” and “d”. The result in this example is strained.

TABLE 1-2: SEARCH PARAMETERS FOR THE COMSOL HELP SYSTEM

OPERATOR	EXAMPLE	SEARCH RESULT EXAMPLE
enclosed quotation marks " "	"time dependent study"	Use quotation marks around a text string to search for exactly that phrase; that is, to search for the words in the order given within the quotation marks.
@	strain@	The "at sign" is a wildcard character. All instances that include strain, for example, are returned.

The About COMSOL Multiphysics Box



COMSOL 4.4

Version number: 4.4.0.126

This product is licensed to:

License number:

This software is furnished under a license agreement. It may be used or copied only under the terms of the license agreement. See http://www.comsol.com/sla. © 1998-2013 COMSOL AB. Protected by U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,219,373; and 8,457,932. Patents pending. Portions of this software are owned by Siemens Product Lifecycle Management Software Inc. © 1986-2013. All Rights Reserved. Portions of this software are owned by Spatial Corp. © 1989-2013. All Rights Reserved. None of the LiveLink™ products are affiliated with, endorsed by, sponsored by, or supported by Autodesk, Inc., Dassault Systèmes S.A., Dassault Systèmes SolidWorks Corp., Microsoft Corp., PTC Inc., Siemens Product Lifecycle Management Software Inc., SpaceClaim Corp., The MathWorks, Inc. or any of their affiliates and/or subsidiaries. COMSOL, COMSOL Multiphysics, Capture the Concept, COMSOL Desktop, and LiveLink are either registered trademarks or trademarks of COMSOL AB. Autodesk, AutoCAD and Inventor are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. Other product or brand names are trademarks or registered trademarks of their respective holders. For a list of such trademark owners, see http://www.comsol.com/tm.

Figure 1-4: The About COMSOL Multiphysics dialog box with the Acknowledgments list.

To open the **About COMSOL Multiphysics** (☰) window:

- For Windows users, select it from the **File** menu.
- For Mac and Linux (cross platform) users, choose it from the **Help** menu.

In addition to copyright and patent information, the **About COMSOL Multiphysics** dialog box has the following information:

- The **Version number**
- The user or company **This product is licensed to**
- The **License number**

Select an option from the list below and then click **Show Information** to open a separate window of the same name containing this information:

- Select **Acknowledgments** to show information about third-party software components, including license notices required by the software component authors.
- Select **License agreement** to show the COMSOL Multiphysics software license agreement.
- Select **Licensed products** to show the licensed COMSOL products, including the number of used licenses and the total number of licenses for each product.

- Select **Patents** to show the patents that the COMSOL program is protected by.
- Select **System information** to show a list of system properties, which can be useful for troubleshooting purposes, for example.
- Click **Show Information** to open an **Acknowledgments** window that displays detailed information about your software installation.
- Click **COMSOL Web Page** to open your web browser on the main COMSOL web page.

You can also get information about the licensed products from the **Licensed and Used Products** window.

The Root Window

Checking for Product Software Updates

COMSOL provides product software updates that improve the software and correct issues found.

To check if a product update is available from the **File** menu, select **Help>Check for Product Updates** ( ).

The program then checks if an update that is applicable, but not yet installed, is available from the COMSOL website.

If an update is available, an **Update** dialog box appears where you click **Download** to download the update directly or click **Browse Update** to open the COMSOL website where you can read about and download the update.

If no updates are available, the **Update** dialog box reports that your COMSOL installation is up-to-date. To check for updates automatically each time that you start COMSOL, select the **Check for updates at launch** check box under **Product updates** on the **Updates** page in the **Preferences** dialog box.

Open The Preferences Dialog Box and click **Updates** to select the **Check for updates at launch** check box to make the program check for updates each time COMSOL is launched.

Typographical Conventions

All documentation uses a set of consistent typographical conventions that make it easier to follow the discussion, understand what you can expect to see on the graphical user interface (GUI), and know which data must be entered into various data-entry fields.

In particular, these conventions are used throughout the documentation:

CONVENTION	EXAMPLE
text highlighted in blue	Click text highlighted in blue to go to other information in the PDF. When you are using the help desk in COMSOL, links to other modules, model examples, and documentation sets also work.
boldface font	A **boldface** font indicates that the given word(s) appear exactly that way on the COMSOL Desktop (or, for toolbar buttons, in the corresponding tooltip). For example, the **Model Builder** window is often referred to, and this is the window that contains the model tree. As another example, the instructions might say to click the **Zoom Extents** button (), and this means that when you hover over the button with your mouse, the same label displays on the COMSOL Desktop.
italic font	An *italic* font indicates the introduction of important terminology. Expect to find an explanation in the same paragraph or in the Glossary. The names of other user guides in the COMSOL documentation set also have an *italic* font.
Forward arrow symbol **>**	The forward arrow symbol **>** is instructing you to select a series of menu items or nodes in a specific order. For example, **Component>Mesh** is equivalent to: Under the **Component** node, click **Mesh**.
`code` (monospace) font	A `code` (monospace) font indicates you are to make a keyboard entry in the physics interface. You might see an instruction such as "Enter (or type) 1.25 in the **Current density** field." The monospace font also is an indication of programming code or a variable name.
Italic *`code`* (monospace) font	An italic *`code`* (monospace) font indicates user inputs and parts of names that can vary or be defined by the user.
Arrow brackets <> following the `code` (monospace) or *`code`* (italic) fonts	The arrow brackets included in, for example, programming examples (after a monospace `code` or an italic *`code`* font) mean that the content in the string can be freely chosen or entered by the user, such as feature tags. For example, `model.geom(`*`<tag>`*`)` where *`<tag>`* is the geometry's tag (an identifier of your choice). When the string is predefined by COMSOL, no bracket is used and this indicates that this is a finite set, such as a feature name.

KEY TO THE GRAPHICS

Throughout the documentation, icons are used to help organize the information. These icons vary in importance, but it is recommended that you read these text boxes.

ICON	NAME	DESCRIPTION
⚠	Caution	A Caution icon indicates that the user should proceed carefully and consider the next steps. It might mean that an action is required, or if the instructions are not followed, that there will be problems with the model solution.
!	Important	An Important icon indicates that the information provided is key to the model building, design, or solution. The information is of higher importance than a note or tip, and the user should endeavor to follow the instructions.
📝	Note	A Note icon indicates that the information can be of use to the user. It is recommended that the user read the text.
💡	Tip	A Tip icon is used to provide information, reminders, shortcuts, suggestions for improving model design, and other information that might be useful.
🔍	See Also	The See Also icon indicates that other useful information is located in the named section. If you are working on line, click the hyperlink to go to the information directly. When the link is outside of the current PDF document, the text indicates this, for example "See The Laminar Flow Interface in the *COMSOL Multiphysics Reference Manual*." Note that if you are in the on-line help, the link works.
▥	An example from the model library	The Model icon is used in the documentation as well as in COMSOL. If you are working on line, click the link to go to the PDF version of the step-by-step instructions. In some cases, a model is only available if you have a license for a specific module. The model library path describes how to find the actual model in COMSOL Multiphysics, for example: If you have the RF Module, see Radar Cross Section: model library path **RF_Module/Scattering_and_RCS/radar_cross_section**
Space Dimension		Another set of icons are also used in the Model Builder—the component space dimension is indicated by 0D ▪, 1D —, 1D axial symmetry ⊥, 2D ●, 2D axial ◓ symmetry, and 3D ■ icons. These icons are also used in the documentation to clearly list the differences to a physics interface, node, or theory section, which are based on space dimension.
Win	Windows	The use of this icon indicates that the information is specific to a Microsoft Windows operating system.
Mac	Mac	The use of this icon indicates that the information is specific to a Mac OS operating system. This may also be referred to as cross-platform when describing how to access a feature or menu on the COMSOL Desktop.
Linux	Linux	The use of this icon indicates that the information is specific to a Linux operating system. This may also be referred to as cross-platform when describing how to access a feature or menu on the COMSOL Desktop.

Overview of the Reference Manual

This *COMSOL Multiphysics Reference Manual* provides comprehensive information about all modeling steps using COMSOL Multiphysics. See the individual module manuals for information specific to a specialized Module (see The COMSOL Modules and Interfacing Options for a list).

> As detailed in the section COMSOL Documentation and Help this information can also be searched from the **Help** system in COMSOL Multiphysics.

TABLE OF CONTENTS, GLOSSARY, AND INDEX

To help you navigate through this guide, see the Contents, Glossary, and Index.

ENVIRONMENT

The COMSOL Modeling Environment chapter provides an overview of the COMSOL modeling environment as controlled by COMSOL Desktop and the tools and windows it provides in the Windows version as well as the cross-platform version. Topics include The COMSOL Desktop, The Model Libraries Window, The Physics Interfaces, Creating a New Model with the Model Wizard, and a key to the icons including links in the Toolbars and Keyboard Shortcuts section.

MODELING

Building a COMSOL Model explains a range of methods and topics including information about the following: details about an introduction to The Model Builder, The Component Node, The Physics Nodes, Selecting Physics, Analyzing Model Convergence and Accuracy, Specifying Model Equation Settings, Boundary Conditions, Using Units, and Numerical Stabilization, and much more.

CUSTOMIZING THE COMSOL DESKTOP

In the chapter Customizing the COMSOL Desktop, the settings are described related to Customizing a Model, changing Preferences Settings, and details about the Advanced Physics Sections.

DEFINITIONS

The Definitions chapter describes the global and local (component) definitions features. Depending on the geometric scope, you add the nodes described in this section to either the Global Definitions node or under the Definitions node for a particular component. Topics include Operators, Functions, and Constants, Predefined and Built-In Variables, Mass Properties, Functions, Component Couplings, Coordinate Systems, Identity and Contact Pairs, Probes, and Infinite Element Domains and Perfectly Matched Layers.

VISUALIZATION AND SELECTION

The Visualization and Selection Tools chapter describes the tools used to visualize and control how you view models and select parts of the model geometry in the Graphics window and the settings windows. Important topics include Working with Geometric Entities, Named Selections, and User-Defined Views.

GEOMETRY

The Geometry Modeling and CAD Tools chapter covers geometry modeling in 1D, 2D, and 3D with examples of solid modeling, boundary modeling, Boolean operators, and other CAD tools in COMSOL. In addition, it shows how to use the tools for exploring geometric properties, such as volumes and surfaces. There is also information about using external CAD data. Topics include Creating a Geometry for Analysis, Working with Geometry Sequences, Geometric Primitives, Geometry Operations, and Virtual Geometry and Mesh Control Operations.

MESH

The Meshing chapter summarizes how to create and control your mesh for 1D, 2D, and 3D geometries in COMSOL. It also explains the possibilities for importing and exporting meshes in different formats. Topics include Creating a Mesh for Analysis, Meshing Techniques, Meshing Operations and Attributes, and Importing and Exporting Meshes.

MATERIAL

The Materials chapter introduces you to the material databases included with COMSOL. Topics include a Materials Overview, Working with Materials, Material Properties Reference, User-Defined Materials and Libraries, Using Functions in Materials, and Module-Specific Material Databases.

AC/DC

The AC/DC Interfaces chapter explains the interfaces available for modeling electromagnetics, which you find under the AC/DC branch ( ) when adding a physics interface. It also contains sections about general fundamentals and theory for electric fields.

ACOUSTICS

The Pressure Acoustics chapter describes how to use the Pressure Acoustics, Frequency Domain interface, found under the Acoustics branch ( ) when adding a physics interface, for modeling and simulation of acoustics and vibrations.

CHEMICAL SPECIES TRANSPORT

The Chemical Species Transport chapter explains how to use the Transport of Diluted Species interface, found under the Chemical Species Transport branch ( ) when adding a physics interface, to model and simulate mass transfer by diffusion and convection based on Fick's law of diffusion.

FLUID FLOW

The Fluid Flow chapter explains how to use the Laminar Flow interface, found under the Fluid Flow>Single-Phase Flow branch ( ) when adding a physics interface, to model and simulate fluid mechanics for laminar, incompressible fluids.

HEAT TRANSFER

The Heat Transfer Modeling chapter describes the different types of Heat Transfer interfaces (Heat Transfer in Solids and Heat Transfer in Fluids), and the Joule Heating multiphysics interface, all found under the Heat Transfer branch ( ) when adding a physics interface.

SOLID MECHANICS

The Structural Mechanics chapter explains how to use the Solid Mechanics interface, found under the Structural Mechanics branch ( ) when adding a physics interface, to simulate and analyze applications involving solid mechanics. The interface is used for stress analysis and general solid mechanics simulation. This chapter also includes a section about Using Load Cases.

EQUATION-BASED MODELING

The Equation-Based Modeling chapter describes the use of the mathematics interfaces, found under the Mathematics branch (Δu) when adding a physics interface, which are used for equation-based modeling. With those interfaces you can solve various types of PDEs using different formulations. You can also solve ODEs and other global equations.

SENSITIVITY ANALYSIS

The Sensitivity Analysis chapter describes how to perform sensitivity analysis using the Sensitivity interface, found under the Mathematics>Optimization and Sensitivity ( ) branch when adding a physics interface.

DEFORMED MESHES

The Deformed Geometry and Moving Mesh chapter explains how to use the modeling interfaces that control mesh deformation. It also contains fundamentals about deformed meshes and information about the Eulerian and Lagrangian formulations of the physics, the frame types that support these formulations, and the arbitrary Lagrangian-Eulerian (ALE) method.

STUDIES AND SOLVERS

The Studies and Solvers chapter lists the various types of solvers and studies in COMSOL and explains the study steps and solver configurations. It also describes the major solvers and settings as well as batch jobs, parametric sweeps, and cluster computing. See also the *Optimization Module Manual* for other supplementary information.

RESULTS AND VISUALIZATION

The Results Analysis and Plots chapter helps you analyze results in COMSOL and describes numerous result-evaluation and visualization tools, including advanced graphics, data display, and export functions. Topics include Postprocessing and Analyzing Results, Results, Data Sets, Plot Groups and Plots, Derived Values and Tables, Exporting Data and Images, Reports, and Printing and Capturing Screen Shots.

RUNNING COMSOL

Running COMSOL is an overview of the different ways that you can run the COMSOL Multiphysics software in addition to running the COMSOL Desktop graphical user interface on a dedicated computer, including client/server and distributed-memory architectures and cloud-based computing.

2

The COMSOL Modeling Environment

The COMSOL Desktop® provides a complete and integrated modeling environment for creating, analyzing, and visualizing multiphysics models. This chapter provides an overview of the COMSOL Multiphysics® modeling environment as controlled by COMSOL Desktop and the tools and windows it provides.

The COMSOL Desktop

This section is an overview of the major components in the COMSOL Multiphysics environment. These components are integrated into the *COMSOL Desktop*, which you can personalize to your own modeling needs and preferences. Primarily consisting of the *Model Builder*, *node settings* windows, and *Graphics* windows, other dockable windows can be opened, closed, and organized according to the modeling settings you need to access and the GUI configuration you want to work in. You can save these configurations, and the last opened configuration is always displayed when you open COMSOL again.

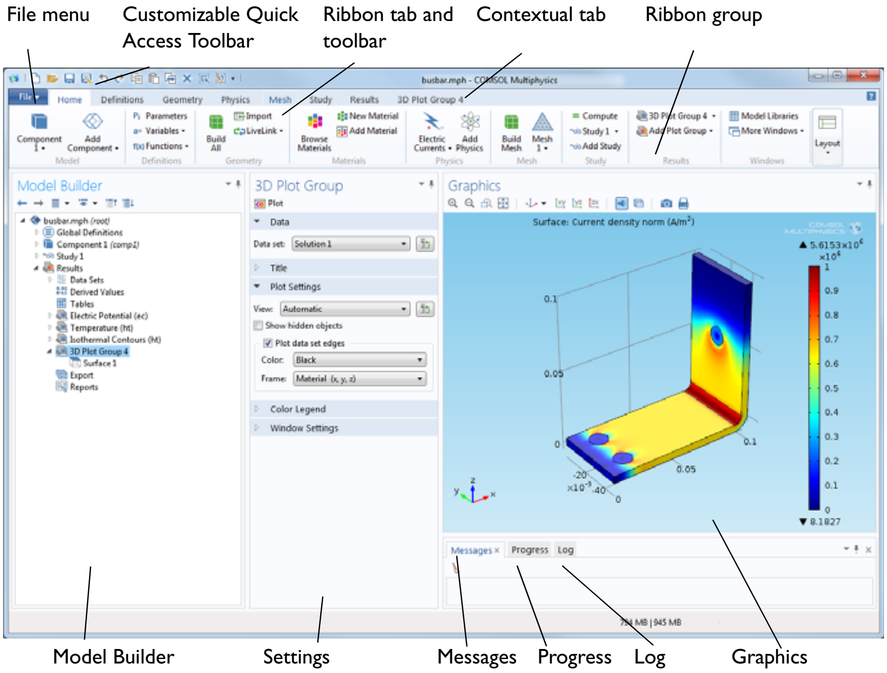
- Creating a New Model
- Building a COMSOL Model
- Customizing the COMSOL Desktop
- The Model Builder

Mac

Linux

The COMSOL Desktop in the cross-platform version, primarily for the Linux and Mac operating systems, looks slightly different than for the Windows operating system, which is shown in Figure 2-1. The primary difference is that the Main Menu and Main Toolbar are used instead of ribbons. Otherwise, the default windows (Model Builder, Graphics, Settings, Log, Progress, and Messages) are in the same location on the default Desktop layout. See The COMSOL Desktop Menus and Toolbars for more details.

You can also launch the cross-platform version on Windows using `comsolxpl.exe`.



Data set: Solution 1

View: Automatic

☐ Show hidden objects

☑ Plot data set edges

Color: Black

Frame: Material (x, y, z)

Figure 2-1: The default COMSOL Desktop with its major windows in a widescreen layout. The ribbon tabs and groups are available for Windows users. For Mac and Linux users the layout is similar but you access some options from the main menu or contextual toolbars.

Win

A *ribbon tab*, *ribbon group*, or *modal ribbon tab*, are available in the Windows version. See Figure 2-1 for an example of the Windows **Home** ribbon. Also see Figure 2-2 for an example of how the ribbon changes when a window is resized.

Mac

Linux

The *Model Toolbar* and *Contextual Toolbar* are available in the cross-platform version, primarily for Mac and Linux users. See Figure 2-9 for an example of these toolbars.

ABOUT CHANGES TO THE RIBBON DISPLAY (WINDOWS USERS)

When the complete COMSOL Desktop is resized, the ribbon toolbar groups are collapsed and grouped under menus instead of displaying individual buttons. In Figure 2-2, all the groups on the Home ribbon are collapsed into menus. As the window is widened, the ribbon groups expand to include the options as buttons or other submenus instead.

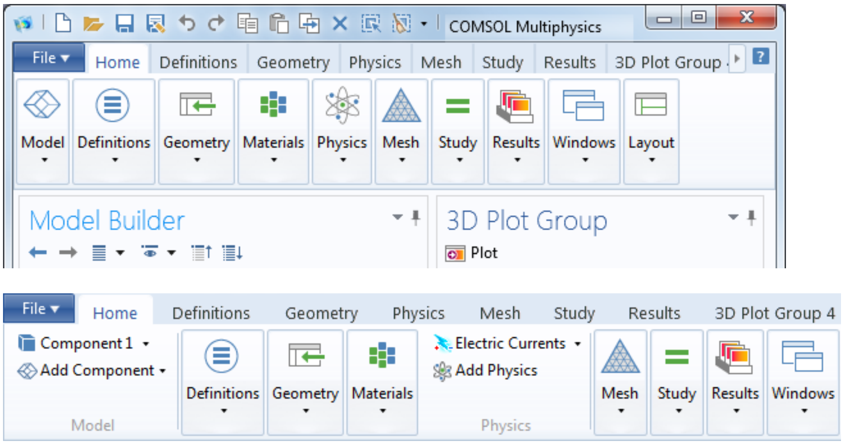


Figure 2-2: When the COMSOL Desktop is resized, the ribbon toolbar buttons are grouped together with the ribbon tab group name. In this example for the Home ribbon, all the buttons are available from a menu, such as Definitions, Geometry, Material, Physics, and so forth (top). As the window is widened, the menus expand accordingly (bottom).

OVERVIEW

The rest of this section introduces you to the features of the COMSOL Desktop, explains some basic navigation, and provides you with an overview of the windows, toolbars, and menus available. In this chapter you will also learn about the model file formats, the options to save files, and the units systems available for modeling.

- Basic Navigation
- Adjusting Window Location and Size on the Desktop
- The COMSOL Desktop Windows
- The COMSOL Desktop Menus and Toolbars
- Windows Toolbars and Menus
- Cross Platform (Mac and Linux) Toolbars and Menus
- Features Available on Toolbars and From Menus
- The Messages Window
- About the COMSOL Model File Formats
- Saving COMSOL Model Files
- Saving and Opening Recovery Files
- The Root Window
- Unit Systems in COMSOL

After this introductory overview, The Model Libraries Window section explains how to work with the model libraries included with COMSOL. The Physics Interfaces section lists the interfaces available with a basic COMSOL Multiphysics license. This prepares you to start creating a new model.

The next section, Creating a New Model, shows you how to use the Model Wizard to begin building a new model by choosing a physics interface and study combination.

The last section, Toolbars and Keyboard Shortcuts, is a quick reference to all the features found on the toolbars. It includes links to the information contained throughout this reference manual.

Basic Navigation

Basic navigation on the COMSOL Desktop extensively involves the nodes in the Model Builder as well as moving between windows and sections on settings windows.

WORKING WITH NODES IN THE MODEL BUILDER

The following methods are available to select nodes, expand and collapse branches, open the settings window, or move up and down the nodes in the model tree:

- Click a node in the Model Builder to highlight it and to open the associated settings window. See The Node Settings Windows.
- Once a node is highlighted, there are many things you can do; for example, you can copy, duplicate, delete, and move some nodes around. See Copying, Pasting, and Duplicating Nodes, Moving Nodes in the Model Builder, and Clearing Sequences and Disabling, Enabling, and Deleting Nodes.
- Right-click a node to open a context menu. See Opening Context Menus and Adding Nodes.
- When a node is highlighted, use the up arrow key on the keyboard to move to the node above; to move to the node below, use the down arrow key.
- To expand a branch to display all nodes in the branch, click the small left-pointing white triangle next to the branch icon in the model tree, or press the right arrow key. To collapse a branch to display only the main branch node, click the small downward-right pointing black triangle next to the branch icon in the model tree, or press the left arrow key. See The Model Builder Toolbar for information about how to collapse or expand all branches.
- A highlighted node is also dynamic and its appearance can change based on where in the modeling process you are. See Dynamic Nodes in the Model Builder for a list of these visual cues.

The COMSOL Desktop Menus and Toolbars

MOVING BETWEEN WINDOWS AND SECTIONS ON THE COMSOL DESKTOP

Keyboard shortcuts are quick ways to navigate between the windows on the COMSOL Desktop and to switch focus between windows and settings window sections:

- Press Ctrl+Tab to switch focus to the next window on the desktop.
- Press Ctrl+Shift+Tab to switch focus to the previous window in the desktop.
- Press Ctrl+Alt+left arrow to switch focus to the **Model Builder** window.
- Press Ctrl+Alt+right arrow to switch focus to the settings window.
- Press Ctrl+Alt+up arrow to switch focus to the previous section in the settings window.
- Press Ctrl+Alt+down arrow to switch focus to the next section in the settings window.

The section Keyboard Shortcuts has additional shortcuts for all operating systems.

- The COMSOL Desktop
- The Model Builder
- Creating a New Model

Adjusting Window Location and Size on the Desktop

MOVING AND RESIZING THE WINDOW

- To move a window, click-and-drag the window tab (the tab is where the window name—**Model Builder**, for example—displays) to where you want it.
- To resize a window, hover your mouse over the window borders until a double arrow displays. Click-and-drag the borders between windows until the layout is how you want it.

At any time, click the **Reset desktop** button.

FLOATING/DETACHING A WINDOW

Win

To detach a window to move and resize it, right-click the window tab and select **Float**. Right-click the window and choose **Dock** to return it to its default location on the Desktop.

Mac

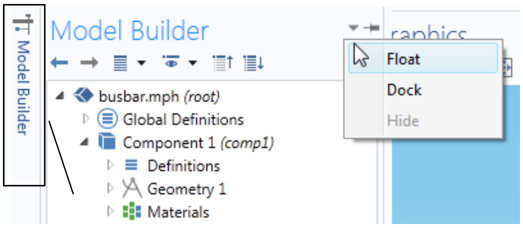
Linux

To detach a window to move and resize it, right-click the window tab and select **Detached**. Right-click and select the option again to dock it to the COMSOL Desktop, or drag and drop it back to where you want it.

HIDING OR PINNING A WINDOW TO THE SIDE OF THE DESKTOP (WINDOWS USERS)

To hide a window, right-click the window and select **Hide**. The window is minimized along the side of the Desktop (see Figure 2-3). Hover over the name to view a hidden/minimized window. To restore a hidden window, either right-click the window, or from the list, select **Float** or **Dock**.

Pinning a window performs the same action as hiding it. Click the pin button in the top-right corner of any window to pin it to the side of the COMSOL Desktop. To return the window to its unpinned state, hover over the window name to open it, then click the pin button (now laying on its side, see Figure 2-3) to restore the window to its default location.



When you hide a window, it is minimized along the side of the Desktop. Hover over the name to view the window.

Figure 2-3: A hidden window is minimized along the side of the Desktop. Hover over the window name to view it. You can then choose to Float or Dock the window (either right-click the window or choose options from the menu), or click the pin icon to restore it to the default location on the Desktop.

USING THE POSITION GUIDES (WINDOWS USERS)

When customizing your COMSOL Desktop, or when you want to return a floating window to the Desktop (Dock it), there are several visual guides available to assist.

Click and hold the mouse on a window to reposition or dock it on the Desktop. This displays the positioning guides (Figure 2-4 and Figure 2-5). Drag the window over any of the guides to highlight the area where the window is to be placed on the desktop (Figure 2-5). The center guide has five options. There are two vertical positioning guides, one on the left and one on the right of the Desktop and two horizontal positioning guides, one on the top and one on the bottom of the Desktop.

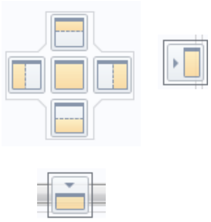


Figure 2-4: Examples of the positioning guides.

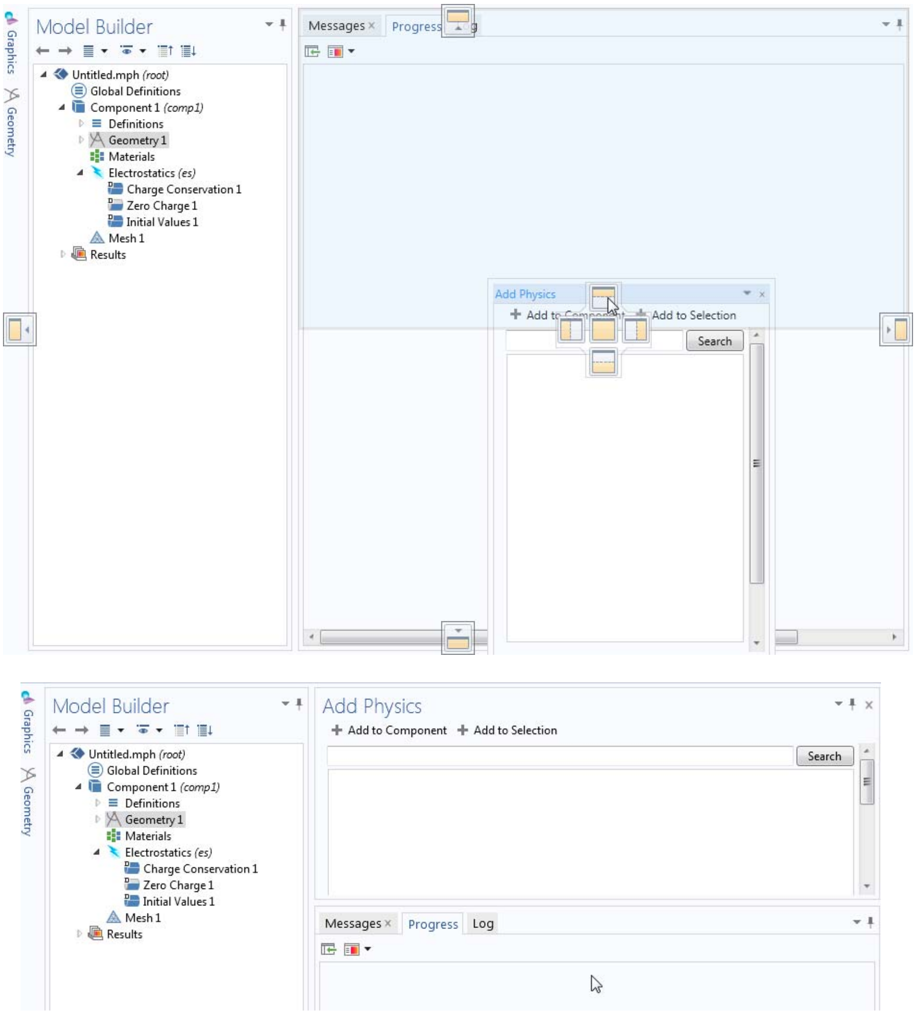


Figure 2-5: The positioning guides display (top image) when you click and hold the mouse pointer on a window. Drag the window over any of the guides to see the highlighted blue area, which indicates the destination for the window. Release the mouse button and the window drops into place (bottom image).

MOVING, MINIMIZING, AND MAXIMIZING WINDOWS (MAC AND LINUX)

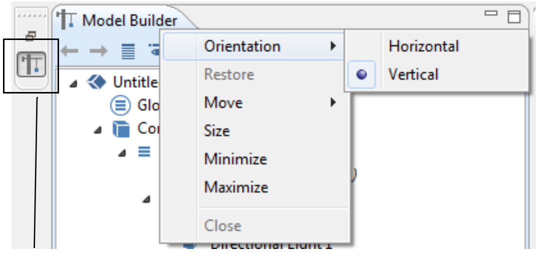
- Right-click the window tab and select **Move>View** (to move a separate window). Move the mouse to where you want the window to display and left-click to confirm the move.
- Right-click the window tab and select **Move>Tab Group** (to move several tabbed windows) from the list. Move the mouse to where you want to the group of windows to display and left-click to confirm the move.
- To resize a window, hover the mouse over the left, right, top, or bottom boundaries of the window until a double arrow displays. Drag the mouse to resize the window. Or right-click the window tab and select **Size>Left**, **Right**, **Top**, or **Bottom**. A blue line highlights the choice; drag to resize.

- To maximize and restore a window’s original position, double-click a window tab to maximize it; double-click again to restore it.
- Click the **Minimize** or **Maximize** button in the top-right corner or right-click the window tab and select **Minimize** or **Maximize** from the list.

At any time, click the **Reset desktop** button on the main toolbar. The section Keyboard Shortcuts has additional shortcuts for all operating systems.

VERTICAL OR HORIZONTAL WINDOW ORIENTATION (MAC AND LINUX)

After a window is minimized along the side of the Desktop, you also have the option to change the window **Orientation** to **Vertical** (the default) or **Horizontal** when you click the window icon (see Figure 2-6).



When you minimize a window, it is available from the side of the Desktop. Click the button to view the window.

Figure 2-6: A minimized window is accessible from the side of the Desktop. Click the window icon to view it. You can then right-click the window to Move, Size, Minimize or Maximize the window. You can also change the Orientation of a minimized window to be Horizontal or Vertical when you click and view it on the Desktop in its minimized state.

- The COMSOL Desktop
- The Model Builder
- Creating a New Model

The COMSOL Desktop Windows

The COMSOL Desktop windows, including those shown in Figure 2-1, are integral to building your model. The windows listed in Table 2-1 are described throughout the documentation and the table includes links to this information.

TABLE 2-1: COMSOL DESKTOP WINDOWS

WINDOW NAME AND LINK	DESCRIPTION
BUILDING A MODEL	
The Model Wizard	Start building a model by choosing the Component space dimension, physics interfaces, and study.
The New window	Open the New window to begin modeling using the Model Wizard to start with a Blank Model. See Open a New Window to Begin Modeling.
The Model Builder	Control the modeling procedure using the model tree. This window has all the functionality and operations for building and solving models and displaying the results.

TABLE 2-1: COMSOL DESKTOP WINDOWS

WINDOW NAME AND LINK	DESCRIPTION
The Graphics Window	This window is a graphical view of the geometry, mesh, and results of the model. It also has useful tools to change the view and select multiple geometric entities, for example.
The Material Browser Window	Browse the material libraries and load materials into your models.
The Add Material Window	Add predefined materials to models.
The Add Physics Window	Add physics interfaces to models.
The Node Settings Windows	When a node is clicked in the Model Builder a corresponding settings window opens with the same name as the node. It is a window with settings that define operations and properties specific to that node.
The Selection List Window	Choose objects, for example, while working with complex geometries and when you need to easily locate a geometric entity that is not easily viewed.
The Measure window	A tool used to measure geometry objects and entities. See Measuring Geometry Objects.
RESULTS AND ANALYSIS	
The Add Study Window	Add a study or studies to models.
The Plot Windows	Plot windows are also graphics windows. These plot windows display convergence results and monitor probe values while solving, for example.
The Table window	Displays the results from integral and variable evaluations defined in Derived Values nodes or by Probes and stored in Table nodes. See The Table Window and Tables Node.
The Messages Window	Contains information useful after an operation is performed.
The Progress Window	Displays the progress of the solver or mesher during the process, including a progress bar and progress information for each solver or mesher.
The Log Window	Contains information from previous solver runs, including convergence information, solution time, and memory use.
The External Process Window	Follow external processes (such as distributed batch jobs) that have been started. The window updates when you are attached to the external processes.
The Mesh Statistics Window	This window includes information about the minimum and average mesh element quality and a mesh element quality histogram, which shows the relative frequency of mesh elements with different quality values.
MODEL EXAMPLES AND DOCUMENTATION	
The Model Libraries Window	Displays all the models included with an installation. The folders contain the models specific to the installed module.
The Model Library Update Window	A service that provides new and updated models for each of the model libraries of the COMSOL products that your license includes.

TABLE 2-1: COMSOL DESKTOP WINDOWS

WINDOW NAME AND LINK	DESCRIPTION
HELP AND DOCUMENTATION	
The Help Window and Topic-Based Help	Provides access to context help in the COMSOL Desktop.
The Documentation Window	Navigate to PDF or HTML versions of the documentation (availability is based on your license), as well as Search, Bookmark, Print Topics, and Link with Contents.
The Root Window	The root node is the top most level of the Model Builder tree. When you click this node, the Root window opens and includes detailed information about the model file.

- Creating a New Model
- The COMSOL Desktop
- COMSOL Documentation and Help
- Toolbars and Keyboard Shortcuts

The COMSOL Desktop Menus and Toolbars

The menus and toolbars available from the COMSOL Desktop vary slightly between operating systems. As described in the section About This Release of COMSOL Multiphysics, the variations are subtle and the overall functionality remains the same.

The sections Windows Toolbars and Menus and Cross Platform (Mac and Linux) Toolbars and Menus show examples of the main terms and locations of the toolbars and menus.

The Model Builder Toolbar is the same for all platforms and is described in this section.

The Features Available on Toolbars and From Menus section details the available features and functions.

THE MODEL BUILDER TOOLBAR

The Model Builder toolbar is the same for all operating systems. It is located at the top of the window as shown in Figure 2-7. The actions listed in Table 2-2 are used to navigate the Model Builder tree, to show and hide various features, and to move back and forth in the tree.

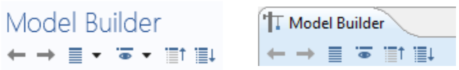


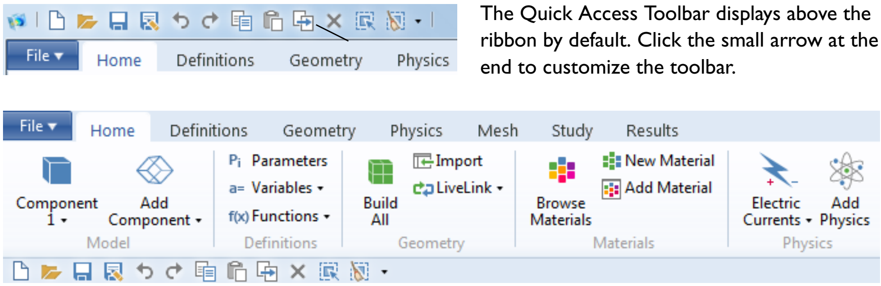
Figure 2-7: The Model Builder toolbar for Windows (left) and Mac and Linux (right). The only difference is in appearance and how options are selected for two of the buttons.

- The COMSOL Desktop
- Creating a New Model
- The Toolbars and Keyboard Shortcuts section has detailed information about the contextual toolbars available on the COMSOL Desktop.

Windows Toolbars and Menus

The available ribbon toolbar options are dynamic, based on where in the model you are working and what is logically available for a specific task. When a blank model is created, only the default tabs are included (Home, Definitions, Study, and Results). The Physics, Geometry, and Mesh tabs are added once a model and physics interface are added to the Model Wizard, as shown in Figure 2-8.

The top of the COMSOL Desktop includes a customizable Quick Access Toolbar. Underneath this are ribbon tabs and ribbon groups, which together, are referred to as *toolbars*, except for the **Home** ribbon, which is a collection of frequently used features from all the other toolbars. For documentation purposes, a toolbar uses the same name as the tab. For example, the **Physics** toolbar, **Geometry** toolbar, or **Study** toolbar. See The Model Builder Toolbar and Features Available on Toolbars and From Menus for a detailed list of all the features available.



The Quick Access Toolbar displays above the ribbon by default. Click the small arrow at the end to customize the toolbar.

Figure 2-8: The Quick Access Toolbar can be positioned above or below the ribbons. You can also customize the toolbar to include or exclude a variety of buttons.

CUSTOMIZE THE QUICK ACCESS TOOLBAR

The Quick Access Toolbar has several default buttons that can be displayed above or below the ribbon. Click the small arrow at the end of the toolbar to open the **Customize the Quick Access Toolbar** list. You can either edit which of the default buttons display directly from the list, or click **More Commands** to **Add** and **Remove** (or double-click to add or remove) the buttons as detailed in the section Features Available on Toolbars and From Menus. This can also be done in the The Preferences Dialog Box in the **Quick Access Toolbar** section.

DISPLAY THE QUICK ACCESS TOOLBAR ABOVE OR BELOW THE RIBBON

Right-click a ribbon to select **Show Quick Access Toolbar Above the Ribbon** or **Show Quick Access Toolbar Below the Ribbon**. These options are also available from the Customize Quick Access Toolbar menu.

Select **Minimize the Ribbon**. To restore the ribbon, right-click anywhere in the top of the window and click **Minimize the Ribbon** to deactivate it (remove the check mark).

MINIMIZE (HIDE) THE RIBBON

Right-click anywhere on a ribbon and choose **Minimize the Ribbon** to hide the ribbon on the Desktop. To access the ribbon features, click the ribbon tab name (for example, **Home**, **Definitions**, or **Study**). The ribbon features are then available. To restore the ribbon to the top of the Desktop, right-click in the tab name area and click to remove the check mark next to **Minimize the Ribbon**.

Cross Platform (Mac and Linux) Toolbars and Menus

For cross platform users (primarily Mac and Linux), the **Main Toolbar** is similar to the Quick Access Toolbar for Windows. In addition, there is a **Model Toolbar** and a variety of **Contextual Toolbars** available. These are a mixture of drop-down menus and buttons for frequently used actions. For documentation purposes, a toolbar uses the same name as the contextual toolbar. For example, the **Physics** toolbar, **Geometry** toolbar, or **Study** toolbar. See The Model Builder Toolbar and Features Available on Toolbars and From Menus for a detailed list of all the features available.

The Contextual Toolbar changes when you click a **Definitions**, **Geometry**, **Mesh**, **Study**, or **Results** node in the **Model Builder**. The Model Toolbar and Contextual Toolbar are similar to the ribbon toolbars for a Windows operating system.

You can also launch the cross-platform version on Windows using `comsolxpl.exe`.

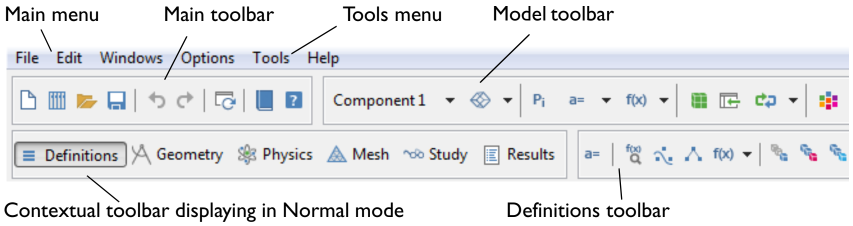


Figure 2-9: The menu and toolbar options for cross platform users (usually Mac and Linux operating systems). Only part of the Model Toolbar and Contextual Toolbars are shown. When one of the buttons is clicked on this toolbar, the associated toolbar opens, in this example for the Definitions node. This toolbar also opens when the Definitions node is clicked in the Model Builder.

DISPLAY OR HIDE THE TOOLBARS FROM THE TOOLS MENU

From the **Tools** menu, you can choose to display or hide each toolbar. Select **Main Toolbar**, **Model Toolbar**, or **Contextual Toolbar** to turn that toolbar on or off in the COMSOL Desktop. For the **Toolbar Button Label**, you can also choose to **Show Icon Only** or **Show Icon and Text**. Finally, choose the **Toolbar Display Mode** as **Normal** or **Compact**. Compact compresses some buttons on the Contextual Toolbar and Model Toolbar under menus instead.

OTHER USEFUL FUNCTIONS AVAILABLE FROM THE WINDOWS MENU

From the **Windows** menu there are also other useful functions:

- Open a variety of useful windows. See The COMSOL Desktop Windows for a list and links to applicable sections.
- From the **Model Builder Node Label** submenu, choose a way to label the nodes in the Model Builder. See Viewing Node Names, Identifiers, Types, and Tags.
- From the **Desktop Layout** submenu, choose a **Widescreen** or **Regular Screen** layout, or **Reset the Desktop**. See Customizing the Desktop Layout.

Features Available on Toolbars and From Menus

The features listed in Table 2-2 are often accessed from multiple locations. In general, the button or menu option is located as follows, with some minor differences between Windows and the cross platform (Mac and Linux) systems and as described in Windows Toolbars and Menus and Cross Platform (Mac and Linux) Toolbars and Menus.

- The **File** menu. See Figure 2-8 (Windows) and Figure 2-9 (Mac and Linux).
- The **Model Builder** toolbar. See Figure 2-7.
- The **Quick Access Toolbar** (Windows only see Figure 2-8). Customize the toolbar to access some of the buttons listed in the table.

- **Main Menu** and a **Main Toolbar** (Mac and Linux, see Figure 2-9).
- Additional menus along the top of the Desktop—**Edit**, **Windows**, **Options**. **Tools**, and **Help** (Mac and Linux, see Figure 2-9).

Cross platform generally implies Mac and Linux users. However, some users may also decide to launch the cross-platform version on Windows using `comsolxpl.exe`.

TABLE 2-2: FEATURES AVAILABLE ON VARIOUS TOOLBARS AND FROM MENUS

ICON	NAME	DESCRIPTION OR LINK TO MORE INFORMATION
Creating Models		
🗋	New	Open the New window to begin modeling using the Model Wizard or start with a Blank Model. Also press Ctrl+N. See Creating a New Model. For all users, this is available from the **File** menu. It is also available on the Quick Access Toolbar (Windows users) or the Main Toolbar (cross platform users). Additional options are available when check boxes are enabled on the Preferences dialog box under Builder Tools.
Opening and Saving Files		
📂	Open	Open an existing model file located on the computer. Also press Ctrl+O. For all users, this is available from the **File** menu. It is also available on the Quick Access Toolbar (Windows users) or the Main Toolbar (cross platform users).
—	Recent and Recent File	From the **File** menu, select a recent model file to open. For Windows users, the file is selected from the **Recent** submenu. For cross platform users, the most recent files are listed at the bottom of the list. Windows users can also customize the Quick Access Toolbar to access this button 📂 .
📚	Model Libraries	Open The Model Libraries Window window. For Windows users, this is available on the **Home** ribbon or from the **File** menu. You can also customize the Quick Access Toolbar and then click the button. For cross platform users, this is available on the Main Toolbar or from the **Windows** menu.
🗘	Open Recovery File	COMSOL can store recovery files each time you start a solver. This is a preference setting that is initially active by default. For all users, this is available from the **File** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Saving and Opening Recovery Files.
💾	Revert to Saved	Opens the last saved version of the file and reinitializes the GUI. For all users, this is available from the **File** menu. It is also available on a customized Quick Access Toolbar (Windows users) or the Main Toolbar (cross platform users). See Reverting to the Last Saved Model File.

TABLE 2-2: FEATURES AVAILABLE ON VARIOUS TOOLBARS AND FROM MENUS

ICON	NAME	DESCRIPTION OR LINK TO MORE INFORMATION
	Save	Save the current model file. Also press Ctrl+S. For all users, this is available from the **File** menu. It is also available on the Quick Access Toolbar (Windows users) or the Main Toolbar (cross platform users). See Saving COMSOL Model Files.
	Save As	Choose to save in one of three COMSOL model file formats. The **Save** dialog box opens and from the **Save as type** list select: **COMSOL Multiphysics Model File (*.mph)** (the default), **Model file for Java (*.java)**, or **Model file for MATLAB (*.m)**. For all users, this is available from the **File** menu. It is also available on the Quick Access Toolbar (Windows users). See About the COMSOL Model File Formats.
Client Server		
	Connect to Server	To connect to a server from the COMSOL Desktop. For all users, this is available from the **File>Client Server** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Connecting to a Server.
	Disconnectfrom Server	To close the connection to the server or MATLAB. For all users, this is available from the **File>Client Server** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Disconnecting from a Server.
	Import Model from Server	To import a particular model when working with MATLAB, Excel, or the COMSOL API. For all users, this is available from the **File>Client Server** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Working with MATLAB, Excel, or the COMSOL API.
	Remove Model from Server	To delete models (remove them from the server) that you have created using `ModelUtil`. For all users, this is available from the **File>Client Server** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Working with MATLAB, Excel, or the COMSOL API.
Model Builder Toolbar		
← →	Previous node Next node	Navigate back to the last node previously selected or to the next node in the sequence. See also Keyboard Shortcuts.
	Expand sections	Click to select an option from the list. See Advanced Physics Sections and Showing and Expanding the Equation Sections and Equation Node.
	Show	Click to select an option from the list. See Advanced Physics Sections and Show More Physics Options.
	Collapse all Expand all	Click to collapse or expand all nodes in the model tree, except the top nodes on the main branch.

TABLE 2-2: FEATURES AVAILABLE ON VARIOUS TOOLBARS AND FROM MENUS

ICON	NAME	DESCRIPTION OR LINK TO MORE INFORMATION
Undo, Redo, Copy, Paste, Duplicate, and Delete		
↶ ↷	Undo Redo	Undo and Redo the last operation for some operations (such as adding, disabling, moving, and deleting nodes in the Model Builder) as well as changing values in the settings window. Also press Ctrl+Z (Undo) or Ctrl+Y (Redo). For Windows users, this is available on the Quick Access Toolbar. For cross platform users, this is available on the Main Toolbar or from the **Edit** menu. See Undoing and Redoing Operations.
	Copy	Copy, paste, and duplicate some features. Also right-click a node to select one of these options from the context menu. For Windows users, this is available on the Quick Access Toolbar. For cross platform users, this is available on the Main Toolbar or from the **Edit** menu. See Copying, Pasting, and Duplicating Nodes.
	Paste	
	Duplicate	
✕	Delete	Delete some nodes while building a model. Also press the Del key or right-click a node to select this option from the context menu. For Windows users, this is available on the Quick Access Toolbar. For cross platform users, this is available on the Main Toolbar or from the **Edit** menu. See Clearing Sequences and Disabling, Enabling, and Deleting Nodes.
	Select All	To select all or clear the selection of all geometric entities in the Graphics window, click the **Select All** or **Clear Selection** buttons, respectively. For Windows users, this is available on the Quick Access Toolbar. For cross platform users, this is available from the **Edit** menu. See Selecting and Clearing Selection of Geometric Entities.
	Clear Selection	
Other		
	Compact History	The model files for Java and for MATLAB contain the entire history of the model, including settings that are no longer part of the model. For all users, this is available from the **File** menu. It is also available on a customized Quick Access Toolbar (Windows users). See Compacting the Model History.
	Reset Desktop	Set the COMSOL Desktop back to widescreen or regular screen, or it reset to default settings. For Windows users, this is available on the **Home** ribbon, **Layout** menu. You can also customize the Quick Access Toolbar and then click the button. For cross platform users, this is available on the Main Toolbar or from the **Windows>Desktop Layout** menu. See Customizing the Desktop Layout.

TABLE 2-2: FEATURES AVAILABLE ON VARIOUS TOOLBARS AND FROM MENUS

ICON	NAME	DESCRIPTION OR LINK TO MORE INFORMATION
🔑	Licensed and Used Products	Open the Licensed and Used Products window to list or block the products your license includes. See Checking and Controlling Products and Licenses Used. For Windows users, this is available from the **File** menu. You can also customize the Quick Access Toolbar and then click the button. For cross platform users, this is available from the **Options** menu.
☰	Preferences	To make changes to how items are displayed or available throughout COMSOL. See Preferences Settings. For Windows users, this is available from the **File** menu. You can also customize the Quick Access Toolbar and then click the button. For cross platform users, this is available from the **Options** menu.
Help and Documentation		
📘	Documentation	Open the Documentation. Also press Ctrl+F1. For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available on the Main Toolbar or from the **Help** menu. See COMSOL Documentation and Help.
?	Help	Open the context Help. Also press F1. See COMSOL Documentation and Help. For Windows users, this is available from the **File>Help** menu or in the upper right-corner of the Desktop. For cross platform users, this is available on the Main Toolbar or from the **Help** menu.
👤	Support Center	Go to the online Support Center on the COMSOL website. For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available from the **Help** menu.
📅	Training	Go to the Training page on the COMSOL website. For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available from the **Help** menu.
⟳	Check for Product Updates	For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available from the **Help** menu. See Checking for Product Software Updates.
⟳	Update the COMSOL Model Library	For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available from the **Help** menu. See The Model Library Update Window.
☰	About COMSOL Multiphysics	For Windows users, this is available from the **File>Help** menu. For cross platform users, this is available from the **Help** menu. See The About COMSOL Multiphysics Box.

The Messages Window

The **Messages** window (✉) displays by default and contains information useful to you after an operation is performed.

The information in this window includes:

- Details about opening and saving models such as MPH-files.
- Information about geometry objects imported from CAD files.
- On the **Mesh** and **Geometry** toolbars, click the **Measure** ( ) button to view information about:
  - The geometry finalization (forming a union or an assembly) and about the number of geometric entities (domains, boundaries, and so on) in the finalized geometry.
  - The number of mesh elements and degrees of freedom in the model.
- Solution times.
- Error messages. The messages are in chronological order and can be scrolled through.

To open the **Messages** window:

- From the **Home** ribbon (Windows users) select **More Windows>Messages**.
- From the main menu (Mac and Linux users) select **Windows>Messages**.
- To clear the window of all messages, click the **Clear** button ( ).

- Meshing
- Geometry Modeling and CAD Tools
- Studies and Solvers

About the COMSOL Model File Formats

There are three COMSOL model file formats: MPH-files, model files for Java, and model files for MATLAB.

THE ROOT NODE

When you first open or create a new model, the *root* node is the topmost level of the tree. By default, unnamed files are called `Untitled.mph`. The file name changes when the file is saved for the first time, but the root node *identifier* does not change for this top level node. See The Root Window for details about the settings available when this node is clicked.

COMSOL MPH-FILES

The default standard file format with the extension `.mph`. The files contain binary and text data. The mesh and solution data are stored as binary data, while all other information is stored as plain text.

You can quickly save and load MPH-files. All the models in the COMSOL Multiphysics model library and the model libraries in the modules are MPH-files.

The MPH-files in the COMSOL model library can have two formats:

- *Full MPH-files* include all meshes and solutions. In the Model Libraries window these models appear with the icon ● . If the MPH-file's size exceeds 25MB, a tooltip with the text "Large file" and the file size appears when you position the cursor at the model's node in the Model Libraries tree.
- *Compact MPH-files* include all settings for the model but have no built meshes and solution data to save space (a few compact MPH-files have no solutions for other reasons). You can open these models to study the settings and to mesh and re-solve the models. It is also possible to download the full versions—with meshes and

solutions—of most of these models through Model Library Update (see The Model Library Update Window). In the Model Libraries window these models appear with the icon . If you position the cursor at a compact model in the Model Libraries window, a **No solutions stored** message appears. If a full MPH-file is available for download, the corresponding node's context menu includes a **Download Full Model** item ( ).

File Locking

Only one user can open and edit an MPH-file at the same time. If you try to open an MPH-file that is already open in another user's COMSOL Desktop, that MPH-file is locked, and you get an option to open the MPH-file in a read-only mode (click **Open As Read-Only**). That means that you can edit the model but you cannot save it unless you save the MPH-file under another name. When an MPH-file is locked, COMSOL creates a separate lock file with the same filename as the MPH-file plus the extension `.lock`, stored in the same directory as the locked MPH-file. If a lock file remains after all COMSOL Desktop sessions have ended (which can happen if the COMSOL Desktop session is ended in a nonstandard way), you can reset the lock when trying to open the file the next time by clicking **Reset Lock and Open**.

Linux and Macintosh do not support operating system locking of files. On those platforms, locking is supported to help users avoid editing the same COMSOL model file, but it is possible to ignore the file locking and delete the lock files.

MODEL FILES FOR JAVA

Editable script files that contain sequences of COMSOL commands as Java code (see the *COMSOL API Reference Manual* for more information about these commands). You can compile these Java files and run them as separate applications. Edit the files in a text editor to add additional commands.

MODEL FILES FOR MATLAB

Model files for MATLAB are editable script files (M-files), similar to the model files for Java, for use with MATLAB. A model file for MATLAB contains a sequence of COMSOL commands as an M-file. You can run these model files in MATLAB like any other M-file scripts. You can also edit the files in a text editor to include additional COMSOL commands or general MATLAB commands.

Running model files in the M-file format requires LiveLink™ *for* MATLAB®.

- The Model Libraries Window
- Saving COMSOL Model Files
- Reverting to the Last Saved Model File
- Printing and Capturing Screen Shots
- Saving and Opening Recovery Files

Saving COMSOL Model Files

The following options are selected from different menus and toolbars as described in The COMSOL Desktop Menus and Toolbars.

SAVING A NEW MODEL

If this is the first time saving a model or you want to update the file and keep the current name and format, in general, these are the ways to save a model:

- Click the **Save** button ( ) on the Quick Access Toolbar or Main Toolbar.
- Press Ctrl+S.
- Select **File>Save**.

CREATING A COPY USING SAVE AS

If the model has been saved before and you want to create a copy you can choose to save in one of three COMSOL model file formats (see COMSOL MPH-Files, Model Files for Java, and Model Files for MATLAB).

Select **File>Save As**. The **Save** dialog box opens and from the **Save as type** list select: **COMSOL Multiphysics Model File (*.mph)** (the default), **Model file for Java (*.java)**, or **Model file for MATLAB (*.m)**.

In all cases, navigate to a the location where you want to save the model, enter a **File name**, and then click **Save**.

You can add the author to the header of model files for Java and for MATLAB that are saved. Open The Preferences Dialog Box and under **General>History export**, select the **Include author** check box.

REVERTING TO THE LAST SAVED MODEL FILE

To open the last saved version of the file and reinitialize the GUI: Select **File>Revert to Saved** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Revert to Saved** ( ) button.

COMPACTING THE MODEL HISTORY

The model files for Java and for MATLAB contain the entire history of the model, including settings that are no longer part of the model. To compact the model history so that the files only include the settings that are part of the current model, select **File>Compact History**. For Windows users, you can also customize the Quick Access Toolbar and then click the **Compact History** ( ) button.

- About the COMSOL Model File Formats
- Windows Toolbars and Menus
- The Root Window
- Printing and Capturing Screen Shots

Saving and Opening Recovery Files

COMSOL can store recovery files each time you start a solver. This setting is initially active by default.

To open a recovery file, select **File>Open Recovery File** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Open Recovery File** ( ) button. See Windows Toolbars and Menus.

The update of the recovery file occurs at the following events:

- After completing the solution for each time step specified as the output times in the **Times** field for a time-dependent simulation.
- After completing each parameter step in a parametric simulation.
- After completing each successful Newton iteration for a nonlinear stationary simulation.

The recovery files are COMSOL MPH-files that represent the state at the time that they were saved. They make it possible to recover from a solver error, which can be especially useful for long time-dependent or parametric runs.

The **Open Recovery File** dialog box lists all recovery files found in chronological order. The files are listed with the date and time when they were saved. When a recovery file is selected, click **OK** to open it in the COMSOL Desktop. Click **Delete** to remove the selected recovery file. Click the **Delete All** button to delete all recovery files.

COMSOL keeps track of the computed time steps or parameter steps in the recovery file, so right-click the **Study** node and select **Continue** ( ) to continue the computation from the point where it was stored in the recovery file. If you are solving a stationary nonparametric problem, the last converged Newton iteration is stored in the recovery file; selecting **Continue** causes the software to resume solving from this stored state.

You can make changes these default settings in The Preferences Dialog Box in the **Temporary Files** section.

- The **Save recovery file** check box is selected by default to save recovery files to disk during the solution process for time-dependent, parametric, and nonlinear solvers. Click to clear as required.
- In the **Folder for recovery files** field, you can specify a different folder from the default to, for example, use a folder where there is more disk space for storing large recovery files. Click **Browse** to browse to a recovery file folder.
- In the **Folder for temporary files** field you can specify a different folder than the default to, for example, use a folder where there is more disk space for storing large temporary files. Click **Browse** to browse to a folder for temporary files.
- If you run COMSOL in a client-server configuration, you can specify a **Folder for recovery files on server**, **Folder for temporary files on client**, and **Folder for temporary files on server**.

Studies and Solvers

The Root Window

The root node is the topmost level of the Model Builder tree. When you click this node, the **Root** node ( ) settings window opens and includes detailed information about the model file. Some of the fields can be edited directly in this window, while others display system information that cannot be changed, or information that changes as updates are made throughout the model (for example, adding a model thumbnail and saving a file with a new name).

When you first open or create a new model, the *root* node is the top most level of the tree. You can change a file name by saving the file, but the root node *identifier, tag*, or *type* cannot be changed for this top level node. This is different than for other nodes in the tree, where these can be edited. See Viewing Node Names, Identifiers, Types, and Tags for information.

NODE PROPERTIES

- **Name**: The file name and file type as determined when the file is saved. See Saving COMSOL Model Files. This field cannot be edited, but the name changes if the file is renamed, for example.
- **Path**: The file location where a file is saved. This field cannot be edited, but is automatically updated when the file is saved to a new path. Also see Documentation and Model Libraries Root Directories.
- **COMSOL version**: Includes the version and build of COMSOL. This is system generated.

- The **Tag** is system generated and at this top level it is `Model`. The number that displays is dependent on how many file models have been opened in the COMSOL instance. That is, every time COMSOL Multiphysics is relaunched, this tag resets to 1.
- **Created** and **Last modified**: These sections cannot be edited and are automatically generated based on the computer system or network time and date settings.
- **Last modified by, Author**: Enter a name of the author of the model.
- **License number**: The license number of the installed software is included here. Also see The About COMSOL Multiphysics Box.
- **Model version**: Enter a tracking version number for the model, for example, `Internal Draft V1`, `Sales Demonstration V2`, or `Version B`.
- **Model description**: Enter any text, for example, a description of the model.

USED PRODUCTS

The information included here is based on the purchased license or modules. See Checking and Controlling Products and Licenses Used. Also see The About COMSOL Multiphysics Box.

MODEL THUMBNAIL

To illustrate the model you can save a model thumbnail image that displays in this section and when opening a model in the **Model Libraries** window. The model thumbnail is a copy of the current plot. See Setting the Model Thumbnail Image for a Model.

UNIT SYSTEM

The default unit system is SI units. Or select any other option from the list. See Unit Systems in COMSOL and Setting the Unit System for Models to change the setting globally or locally.

FONT

The default is to use a predefined **default** font with a font **Size** of 9 points. Depending on the operating system and the installed fonts on the computer, you can select from a number of other font families. See Changing the Font for Plot Labels and Titles to make global changes.

Unit Systems in COMSOL

COMSOL supports the following unit systems:

METRIC UNIT SYSTEMS

- SI units, the International System of Units (SI, Système International d'Unités). This is the default unit system (sometimes also called MKS). For a list of SI units in COMSOL, see SI Base, Derived, and Other Units.
- CGSA units. The CGS system uses centimeter, gram, and second as basic units of length, mass, and time, respectively. The remaining basic units are identical to the SI units. The CGS unit system gives nice values for small lengths, masses, forces, pressures, and energies when working on a microscale and with weak electromagnetic forces. The derived units of force, pressure, and energy have well-known and widely used names: dyne, barye, and erg, respectively. CGSA adds *ampere* as the basic unit for electric current. For a list of CGSA units, see Special CGSA Units.
- Electromagnetic units (EMU). This system is based on Ampère's law, which defines the unit of electric current once you select an appropriate value for the constant C. When dealing exclusively with magnetic effects, it is convenient to set $C = 1$. If CGS units are used for the remaining basic dimensions, the current unit is called an *abampere*, and the corresponding coherent unit system is called electromagnetic units. Unique names for derived units have been introduced by prefixing the SI name with *ab-*. For a list of EMU units, see Special EMU Units.

- Electrostatic units (ESU). Based on Coulomb's law for the force between point charges, ESU uses a unit of charge called the *statcoulomb* with CGS units for length, mass, and time. From there, the *statampere*, or *franklin*, and other derived units of the electrostatic unit system follow. For a list of ESU units, see Special ESU Units.
- MPa units. For stationary structural mechanics, where the density does not appear in the equations, it can be convenient to use a system where newton and megapascal (hence the name "MPa system") are naturally derived units of force and pressure, respectively. Keeping the SI unit for time, the basic units of length and mass become millimeter and tonne. Except for the force and pressure units, other derived units are nameless. For a list of MPa units, see Special MPa Units.

ENGLISH UNIT SYSTEMS

- Foot-pound-second unit system (FPS units). The original foot-pound-second system seems to be the absolute system using the pound as a unit of mass. This version of the FPS system is in agreement with the IEEE standard (the pound is a unit of mass and not of force). The natural derived unit of force is the *poundal*. For a list of FPS units, see Special FPS Units.
- British engineering units. An alternative to the standard FPS system is the British engineering unit system (also called gravitational foot-pound-second system or foot-slug-second system). Here, the pound force is the natural unit of force, which causes the introduction of the mass unit *slug* such that a pound force is a slug-foot per second squared. For a list of British engineering units, see Special British Engineering Units.
- Inch-pound-second unit system (IPS units). It is possible to define varieties of the FPS and British engineering systems based on the inch instead of the foot as basic unit of length. This gives rise to two distinct inch-pound-second systems: the *absolute IPS system* (just called IPS) and the *gravitational IPS system*. For a list of IPS units, see Special IPS Units.
- Gravitational IPS units. This alternative IPS unit system considers the pound a unit of weight rather than a unit of mass. For a list of Gravitational IPS units, see Special Gravitational IPS Units.

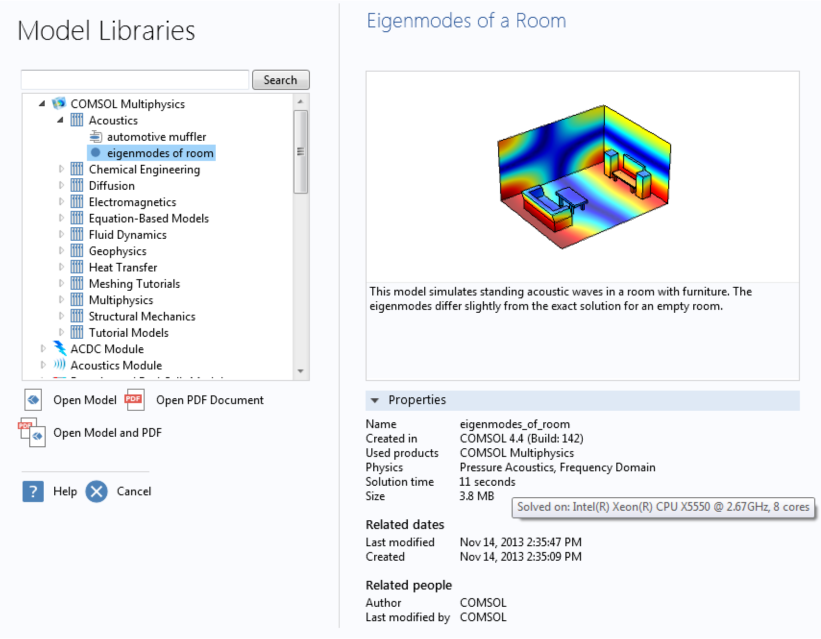
OTHER

- None. No units appear in the settings, which can be useful in nondimensionalized (de-dimensionalized or dimensionless) models.

- Using Units
- Setting the Unit System for Models

The Model Libraries Window

The **Model Libraries** window (Figure 2-10) contains sets of models that can be used for a variety of purposes. Each add-on module includes its own model library with models showing how to use the module within its application areas. Each model includes full model documentation and a brief model description, including the solution times and information about the computer used for solving the model.



Eigenmodes of a Room

This model simulates standing acoustic waves in a room with furniture. The eigenmodes differ slightly from the exact solution for an empty room.

Name	eigenmodes_of_room
Created in	COMSOL 4.4 (Build: 142)
Used products	COMSOL Multiphysics
Physics	Pressure Acoustics, Frequency Domain
Solution time	11 seconds
Size	3.8 MB

Solved on: Intel(R) Xeon(R) CPU X5550 @ 2.67GHz, 8 cores

Related dates

Last modified	Nov 14, 2013 2:35:47 PM
Created	Nov 14, 2013 2:35:09 PM

Related people

Author	COMSOL
Last modified by	COMSOL

Figure 2-10: The Model Libraries window with Eigenmodes of a Room highlighted in the model tree. Specific information about the model is displayed to the right, including the model name, its size, and the solution time.

Browse through the Model Libraries tree to see what models are available for your license. Click to highlight the model in the tree and read the information about it to the right or search for a specific model.

Win

To open the **Model Libraries** window:

- From the **Home** ribbon, click **Model Libraries** ( ). When a toolbar is compressed, you sometimes select it from the **Windows>** menu.
- You can also customize the Quick Access Toolbar and then click the **Model Libraries** ( ) button on the toolbar. See Windows Toolbars and Menus.
- Select **Model Libraries** from the **File** menu.

Mac

Linux

To open the **Model Libraries** window:

- On the Main Toolbar, click the **Model Libraries** ( ) button.
- Select **Windows>Model Libraries**.

The following sections describe what you can do from the Model Libraries window:

- Opening a Model and its Documentation
- Downloading a Full Model
- Searching the Model Libraries
- The Model Library Update Window

You can also set the root directory and create and remove a user-defined model library using the The Preferences Dialog Box as described next.

MODEL LIBRARIES PREFERENCES

Open The Preferences Dialog Box and click **Model Libraries** to edit the following. The buttons are located at the bottom of the **Model Libraries folders** tree.

Add User Model Library

Click the **Add User Model Library** button ( ) to add customized folders. In the **User Model Library** tree in the **Browse for Folder** dialog box, navigate to a location on your computer where you want to create a custom model library folder. Or click **Make New Folder**. Click **Apply** to save the changes, or **Cancel** to exit without saving.

> It is not possible to add a library identical to, containing, or being contained in, any already used library.

Set the Model Library Root

Click the **Set Model Library Root** button ( ) to edit or set the root folder. This redirects COMSOL to a different folder where customized models can be stored.

In the **Model Library Root** tree in the **Browse for Folder** dialog box, navigate to the new root folder location or click **Make New Folder**. Click **Apply** to save the changes, or **Cancel** to exit without saving.

Remove Selected Model Library

This button is enabled after a User Model Library folder is created. Click any model or folder and then the click **Remove Selected** ( ) button to remove it from the Model Library.

Opening a Model and its Documentation

> There are two model formats to choose from in the tree, full and compact. Some models might have been delivered with an MPH-file that contains no stored meshes or solutions. Such compact models are indicated by the icon . See Downloading a Full Model.

OPENING A MODEL IN COMSOL MULTIPHYSICS

Once you have located the model you want to open, for example you used a search and it is successful (see Searching the Model Libraries), or you browsed the Model Libraries tree, to open the model:

- Double-click the model name in the tree.
- Select the model name, then click ( ) **Open Model**.
- Right-click the model name, then from the context menu select **Open Model**.

OPENING A MODEL PDF DOCUMENT

To read the model documentation in PDF format, including step-by-step modeling instructions:

- Click to highlight the model name in the tree, then ( ) **Open PDF Document**.
- Right-click the model name, then from the context menu select ( ) **Open PDF Document**.

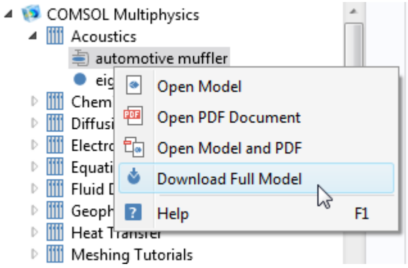
OPENING A MODEL AND ITS PDF DOCUMENT SIMULTANEOUSLY

To open a model and its model documentation PDF document in a single operation:

- Click to highlight the model name in the tree, then ( ) **Open Model and PDF**.
- Right-click the model name, then from the context menu select ( ) **Open Model and PDF**.

Downloading a Full Model

You can download the complete MPH-files via **Model Library Update** (see The Model Library Update Window), or right-click a compact model node ( ) in the **Model Libraries** tree and choose **Download Full Model**.



You can also generate the complete models by building the mesh sequences and computing the studies.

The procedure for restoring the solutions of a model can involve other steps, such as adjusting physics settings. See the individual model documentation for details if the simple approach does not work.

Searching the Model Libraries

You can **Search** the model libraries to find any models using a specific feature. For example, enter all or part of the model name, a physics interface name, a feature name, a feature tag name, or identifier prefixed by '`@`'. or any other phrase or words or and click **Search**.

By default, COMSOL searches for all words in the **Search** field.

COMSOL models are named using an underscore between words (for example, *`effective_diffusivity`*) because the model name is also the name of the corresponding Model MPH-file. The underscore is required to form a valid filename, so it is recommended that you, if you are not sure of the full name, enter only the first word in the **Search** field when searching for a model name.

SEARCH PARAMETERS

- To search for models by filename only, use the prefix “@name:”, for example `@name:busbar`. You can also use the wildcard character ‘*’ at the beginning and the end of the search expression, for example `@name:fluid*` or `@name:*electr*`.
- To search for a phrase, enclose it in quotation marks (for example, “`plane stress`”). The words can be part of the model’s name or its documentation.
- Limit the search to tag names and identifiers with the prefix ‘@’, for example, `@genext` or `@ehs`.
- To search for a specific physics interface, use the scoping syntax `@physics:`*`<identifier>`*. For example, enter `@physics:solid` to find all models that use the Solid Mechanics interface.
- In addition to `@physics`, the supported scopes include `@geom`, `@mesh`, `@probe`, `@result`, `@selection`, and `@study`.

If the search does not return any results, the Model Libraries window contains the message **No Matching Model Found**. Click the **Refresh** button (⟳) to return to the root **Model Libraries** folder list.

If you have any feedback or suggestions for additional models for the libraries (including models you developed), feel free to contact COMSOL at info@comsol.com.

The Model Library Update Window

Using the **Model Library Update** service requires Internet access. For a default installation, you also need to run COMSOL as an administrator. See Proxy Server Settings for Communication with COMSOL's Model Library Update Server section for instructions on how you can modify your installation to avoid this restriction.

Update COMSOL Model Library is a service that provides new and updated models for the model libraries of the COMSOL products that your license includes.

To open the **Model Library Update** window from the **File>Help** menu (Windows users) or from the **Help** menu (Mac and Linux users), select **Update COMSOL Model Library** ( ).

When the Model Library Update window ( ) opens, click to select **Full models** or **Compact models**. Click **Find Models** to check to see if all these models are up-to-date. If the message **Your Model Library is up to date** displays, no updated or new models are available.

If the library is not up-to-date, browse the list of new and updated models that appear with a description and image. Choose which models to download by selecting or clearing the check boxes next to the model thumbnail images. By default all check boxes are selected; by clicking **Uncheck all** and **Check all** you can change the global selection state.

Click the **Download Selected** button to download the selected models, or click the **Download All** button to download all available models. The download time depends on the size of the files, which is listed for each model, and the bandwidth of the Internet connection.

PROXY SERVER SETTINGS FOR COMMUNICATION WITH COMSOL'S MODEL LIBRARY UPDATE SERVER

To edit these settings, open The Preferences Dialog Box and go to the **Updates** section.

If you connect to the Internet through a web proxy, you can use the controls in this section to specify the proxy server settings to use when communicating with the COMSOL website for performing a Model Library Update.

The **Configuration** list has the following options:

- **No proxy server**: Connect to the Model Library Update server directly, bypassing any proxies. This is the default setting.
- **Use system settings**: Use the system-wide proxy server settings defined on your computer.
- **Manual**: Choose this alternative if you want to specify a proxy server by entering the name (or IP address) and port number in the **Server** and **Port number** fields. The default port number is 443, which is the default for HTTP secure (HTTPS). If the proxy server requires authentication, you are asked to provide username and password the first time you run Model Library Update in each COMSOL session.

PLACE MODEL LIBRARY UPDATE FILES IN

To edit these settings, open The Preferences Dialog Box and go to the **Updates** section.

The **Destinations** list provides two options for specifying which model and documentation directories are synchronized with the COMSOL server when you launch a Model Library Update request:

- **Current model and documentation directories** (default): Synchronize with MPH-files under the Model Library root set in the **Model Libraries** window and with model documentation files under the directory specified in the **Documentation root directory** field on the **Preferences** dialog's **General** page.
- **Specify custom directories**: Choosing this option lets you specify model and documentation root directories separate from those of your current COMSOL Desktop environment.

By default, both the model library and documentation root directories lie directly under the COMSOL installation root directory, in `models/` and `doc/`, respectively. This typically implies that special permissions are required for saving downloaded files, and it can therefore be beneficial to move or copy the directories to a different location. The settings referred to in this section are provided to let you customize Model Library Update to the IT environment of your organization.

The Physics Interfaces

This section is an overview of the core physics interfaces included with a COMSOL Multiphysics license. If you have an add-on module, there are additional physics interfaces described in the individual documentation.

- Building Models in the Model Builder
- The Add Physics Window
- Modeling Guidelines
- Creating a New Model

Introduction to the Physics Interfaces

Solving PDEs generally means you must take the time to set up the underlying equations, material properties, and boundary conditions for a given problem. COMSOL Multiphysics, however, relieves you of much of this work. The software provides a number of *physics interfaces* that consist of nodes and settings that set up the equations and variables for specific areas of physics. An extensive set of physics-dependent variables makes it easy to visualize and evaluate the important physical quantities using conventional terminology and notation.

Suites of physics interfaces that are optimized for specific disciplines together with specialized model libraries are available in a group of optional products. See The COMSOL Modules and Interfacing Options.

A complement to the interfaces for physics, special interfaces for equation-based modeling simplify the setup of PDEs for modeling that does not explicitly refer to any particular application field. In addition, other interfaces supplement the physics with special functionality such as the Sensitivity and Moving Mesh user interfaces.

- Physics Groups
- Physics Guide
- Selecting Physics

Physics Groups

The Select Physics page in the **Model Wizard**, as well as The Add Physics Window, contain the main groups of physics and mathematics interfaces as in Table 2-3(some items only display if a license includes the add-on modules).

TABLE 2-3: PHYSICS GROUPS

ICON	BRANCH NAME	DESCRIPTION OF PHYSICS IN THE BRANCH
🕒	Recently Used	Contains the most recently used interfaces for easy access.
⚡	AC/DC	Low-frequency electromagnetics such as electrostatics and electric currents.
)))	Acoustics	Acoustics.
⁘	Chemical Species Transport	Chemical species transport by, for example, convection and diffusion, solving for the species concentrations.

TABLE 2-3: PHYSICS GROUPS

ICON	BRANCH NAME	DESCRIPTION OF PHYSICS IN THE BRANCH
	Electrochemistry	Electrochemistry and modeling of electrochemical components such as batteries and fuel cells. This branch is only available if a license includes the Batteries & Fuel Cells Module, the Electrodeposition Module, or the Corrosion Module.
	Fluid Flow	Fluid flow such as laminar single-phase flow and, with add-on modules, multiphase flow and turbulent flow.
	Heat Transfer	Heat transfer in solids and fluids and thermal multiphysics applications such as Joule heating.
	Optics	Physics interfaces for electromagnetic wave propagation in linear and nonlinear optical media for accurate component simulation and design optimization. This branch is only available if a license includes the Wave Optics Module.
	Plasma	Plasma modeling. This branch is only available if a license includes the Plasma Module.
	Radio Frequency	High-frequency electromagnetic field simulations solving the full Maxwell equations. This branch is only available if a license includes the RF Module.
	Structural Mechanics	Structural mechanics, studying displacements and stresses in solids, for example.
	Semiconductor	The Semiconductor interface solves Poisson's equation for the electric potential and the drift-diffusion equations for electrons and holes in a semiconductor material.
Δu	Mathematics	Mathematics interfaces for solving PDEs, ODEs, and DAEs, for optimization (requires the Optimization Module) and sensitivity analysis, and for modeling moving meshes and parameterized geometry.

- Creating a New Model
- Physics Guide
- Selecting Physics

The Add Physics Window

The **Add Physics** window is similar to the **Select Physics** page accessed through The Model Wizard. It has the same physics interfaces available. This window is a quick way to add physics to models.

To open the **Add Physics** window:

Win

- Right-click a **Component** node and choose **Add Physics**.
- From the **Home** ribbon, click **Add Physics**.
- Select **More Windows>Add Physics**.
- From the **Physics** toolbar, click **Add Physics**.

Mac

Linux

- Right-click a **Component** node and choose **Add Physics**.
- On the **Model Toolbar** click **Add Physics** .
- On the **Physics Contextual Toolbar**, click **Add Physics** .
- Select **Windows>Add Physics**.

The **Add Physics** toolbar button is a toggle button: Click it again to close the **Add Physics** window.

TO ADD PHYSICS TO A MODEL COMPONENT

1 In the **Add Physics** window, either enter a **Search** term or navigate the tree to locate the physics interface to be added to the **Component**.

The tree organizes the available physics interfaces by application areas such as fluid flow, heat transfer, and structural mechanics. The physics found in the modules your license supports display in the different application areas. In some cases, licensing of a module adds physics interfaces to these application areas as well as attributes to existing physics interfaces, which are enhanced with additional functionality. The **Recently Used** branch lists the last five physics interfaces used in recent modeling sessions. You can also enter a text string in the search field and click the **Search** button to list all the interfaces with the search term.

The contents of the **Add Physics** window depends on the space dimension of the active model component. If there are no **Component** nodes in the model, the list of physics interfaces is empty.

2 Once a physics interface is clicked, review and optionally modify any dependent variable names in the **Dependent variables** section and, for some physics interfaces, specify the number of dependent variables.

3 When there is already a physics interface added to the Component, the existing **Studies** are listed under **Physics in study**. Studies that are included appear with a check mark (✓) in the **Solve** column. Click in the column and row to change the check mark to an (✗).

4 Click the **Add to Component** or **Add to Selection** buttons.

- If **Add to Component** is selected, the physics interface is added to the **Model Builder** and becomes active in the entire model component's geometry by default.
- If **Add to Selection** is chosen, the physics is added to the selected geometric entities in the Graphics window and a new node is added to the Model Builder. This is a method called preselection.

- Creating a New Model
- Physics Guide

Physics Guide

The table lists the physics in COMSOL Multiphysics and the availability for 1D, 1D axisymmetric, 2D, 2D axisymmetric, and 3D geometries.

INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
AC/DC				
Electric Currents		ec	all dimensions	stationary
Electrostatics		es	all dimensions	stationary; time dependent
Magnetic Fields		mf	2D, 2D axisymmetric	stationary; frequency domain
Acoustics				
Pressure Acoustics				
Pressure Acoustics, Frequency Domain		acpr	all dimensions	eigenfrequency; frequency domain
Chemical Species Transport				
Transport of Diluted Species		chds	all dimensions	stationary; time dependent
Fluid Flow				
Single-Phase Flow				
Laminar Flow		spf	3D, 2D, 2D axisymmetric	stationary; time dependent
Heat Transfer				
Heat Transfer in Solids		ht	all dimensions	stationary; time dependent
Heat Transfer in Fluids		ht	all dimensions	stationary; time dependent
Electromagnetic Heating				
Joule Heating[1]		—	all dimensions	stationary; time dependent;
Structural Mechanics				
Solid Mechanics		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent

INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Δu **Mathematics**				
Wall Distance		wd	all dimensions	stationary; time dependent
Curvilinear Coordinates		cc	all dimensions	stationary; eigenvalue
Δu *PDE Interfaces*				
Coefficient Form PDE	Δu	c	all dimensions	stationary; eigenvalue; time dependent
General Form PDE	Δu	g	all dimensions	stationary; eigenvalue; time dependent
Wave Form PDE	∫du	wahw	all dimensions	time dependent
Weak Form PDE	∫du	w	all dimensions	stationary; eigenvalue; time dependent
Δu *Lower Dimensions*				
Coefficient Form Boundary PDE	Δu	cb	all dimensions	stationary; eigenvalue; time dependent
Coefficient Form Edge PDE	Δu	ce	3D	stationary; eigenvalue; time dependent
Coefficient Form Point PDE	Δu	cp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent
General Form Boundary PDE	Δu	gb	all dimensions	stationary; eigenvalue; time dependent
General Form Edge PDE	Δu	ge	3D	stationary; eigenvalue; time dependent
General Form Point PDE	Δu	gp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent
Weak Form Boundary PDE	∫du	wb	all dimensions	stationary; eigenvalue; time dependent
Weak Form Edge PDE	∫du	we	3D	stationary; eigenvalue; time dependent
Weak Form Point PDE	∫du	wp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent
d/dt *ODE and DAE Interfaces*				
Global ODEs and DAEs	d/dt	ge	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
Domain ODEs and DAEs	d/dt	dode	all dimensions	stationary; time dependent; eigenvalue

INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Events		ev	all dimensions	time dependent
Boundary ODEs and DAEs	$\frac{d}{dt}$	bode	all dimensions	stationary; time dependent; eigenvalue
Edge ODEs and DAEs	$\frac{d}{dt}$	eode	3D	stationary; time dependent; eigenvalue
Point ODEs and DAEs	$\frac{d}{dt}$	pode	3D, 2D, 2D axisymmetric	stationary; time dependent; eigenvalue
Optimization and Sensitivity				
Optimization Requires the Optimization Module		opt	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
Sensitivity		sens	all dimensions	stationary; eigenfrequency; frequency domain; eigenvalue; time dependent (available with the Optimization Module)
∇^2 *Classical PDEs*				
Laplace Equation	∇^2	lpeq	all dimensions	stationary
Poisson’s Equation	∇^2	poeq	all dimensions	stationary
Wave Equation	∇^2	waeq	all dimensions	time dependent
Helmholtz Equation	∇^2	hzeq	all dimensions	stationary
Heat Equation	∇^2	hteq	all dimensions	stationary; time dependent
Convection-Diffusion Equation	∇^2	cdeq	all dimensions	stationary; time dependent
Deformed Mesh				
Deformed Geometry		dg	all dimensions	stationary; time dependent; frequency domain; eigenvalue
Moving Mesh		ale	all dimensions	stationary; time dependent; frequency domain; eigenvalue

[1] This physics interface is a predefined multiphysics coupling that automatically adds all the physics interfaces and coupling features required.

PHYSICS NODES ADVANCED SETTINGS

Various settings are available by clicking the **Show** button (👁) and selecting from the list of sections to display. See Advanced Physics Sections for more information.

- Creating a New Model
- The Add Physics Window
- Show More Physics Options
- Selecting Physics

Creating a New Model

This section describes how to create a new model using The Model Wizard or to begin with a blank model. First you need to Open a New Window to Begin Modeling. It is also useful have a basic model added to the Model Builder; then you can experiment with The Model Builder, which is described in the Building a COMSOL Model chapter.

Open a New Window to Begin Modeling

To open a **New** window:

- On the **Quick Access Toolbar** (Windows users) or the **Model Toolbar** (Mac and Linux users), click the **New** button ( ).
- Press Ctrl+N.
- Select **File>New**.

After the **New** window opens to the **Model** page, select an option:

- Click the **Model Wizard** button (.mph) to open the **Select Space Dimension** window. Go to The Model Wizard section to continue.
- Click the **Blank Model** button (.mph) to open COMSOL Multiphysics without any model set up in the Model Builder or return to the default COMSOL Desktop. You can then add components and physics to the model.

To enable the **Physics Builder** choose **Preferences** ( ) from the **File** menu (Windows users) or from the **Options** menu (Mac and Linux users). Then under **Builder Tools** select the check box. After the applicable check box is selected under Preferences, under **Physics** click **Physics Builder** (.mphphb) to create custom physics interfaces using the Physics Builder. See the *Physics Builder Manual* for information.

The Model Wizard

The Model Wizard helps you build a model by choosing the space dimension, physics interfaces, and the study you want to use. In the Model Wizard you Select Space Dimension, Select Physics, Review Physics, and finally Select Study.

SELECT SPACE DIMENSION

1 Open the Model Wizard (see Open a New Window to Begin Modeling).

2 On the **Select Space Dimension** page, click to choose the **Component** geometry dimension—**3D**, **2D axisymmetric**, **2D**, **1D axisymmetric**, **1D**, or **0D**.

Component Nodes by Space Dimension

The **Component** node has different icons based on space dimension **0D** (·) (no space dimension), **1D** (—), **1D axisymmetric** ( ), **2D** ( ), **2D axisymmetric** ( ), and **3D** ( ).

0D is used for physics interfaces modeling spatially homogeneous systems such as chemical reacting systems, electrical circuits, and general ODEs. If you want to import a geometry, this is done in the Model Builder but make sure you choose spatial dimensions that this geometry exists in. Remember, not all physics are available for all space dimensions.

Also add a **Component** node to the **Model Builder**:

- By right-clicking the **Root** node ( ) and selecting it from the **Add Component** menu.
- On the **Home** ribbon (Windows users) or the Model Toolbar (Mac and Linux users), select an option from **Add Component** list.

SELECT PHYSICS

On the **Select Physics** page, there are different ways to select one or several physics interfaces to add to the model. There are also interfaces for PDEs, ODEs, and DAEs under the **Mathematics** branch.

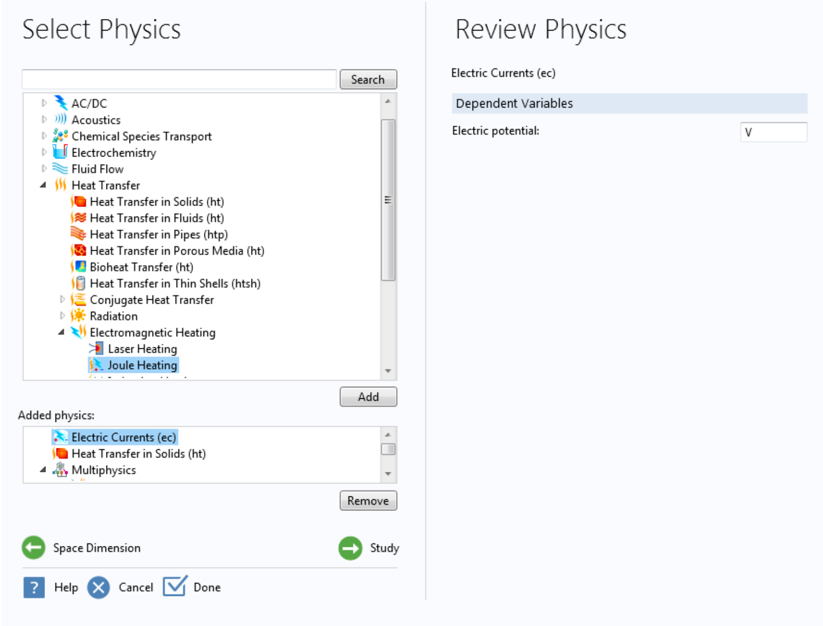
The tree organizes the available physics interfaces on by application areas such as fluid flow, heat transfer, and structural mechanics. The physics found in the modules your license supports display in the different application areas. In some cases, licensing of a module adds physics interfaces to these application areas as well as attributes to existing physics interfaces, which are enhanced with additional functionality. The **Recently Used** branch lists the last five interfaces used in recent modeling sessions. You can also enter a text string in the search field and click **Search** to list all the interfaces with the term.

Once an interface is chosen, there are several options to continue. See Figure 2-11. Click to select one of these buttons:

- **Add** to add the physics interface to the **Added physics** list, or right-click and select **Add physics**. Add as many interfaces as you want. You can use the **Review Physics** page to edit the Dependent Variables as required. Click **Remove** as required to organize the physics interfaces in the list.
- **Space Dimension** to go back to the **Select Space Dimension** page.
- **Study** to choose the study for the model.
- **Done** to add the physics interface without a study.
- **Cancel** to return to the COMSOL Desktop.
- **Help** to open the context-based **Help** window.

Also add physics interfaces from the **Model Builder** and The Add Physics Window:

- On the **Home** ribbon (Windows users) or the **Model Toolbar** (Mac and Linux users), click the **Add Physics** button.
- By right-clicking the **Component** node and selecting **Add Physics**.



Electric potential: V

Figure 2-11: The Select Physics and Review Physics pages in the Model Wizard.

REVIEW PHYSICS

Under **Added physics** (see Figure 2-11), click any interface to open the **Review Physics** page. Here you can review and optionally modify any **Dependent Variables** names and, for some interfaces, specify the number of dependent variables. For other interfaces you can edit both the name of the field and the field components. Examples of fields with components are the displacement field in a **Solid Mechanics** interface and the **Velocity field** for a **Laminar Flow** interface.

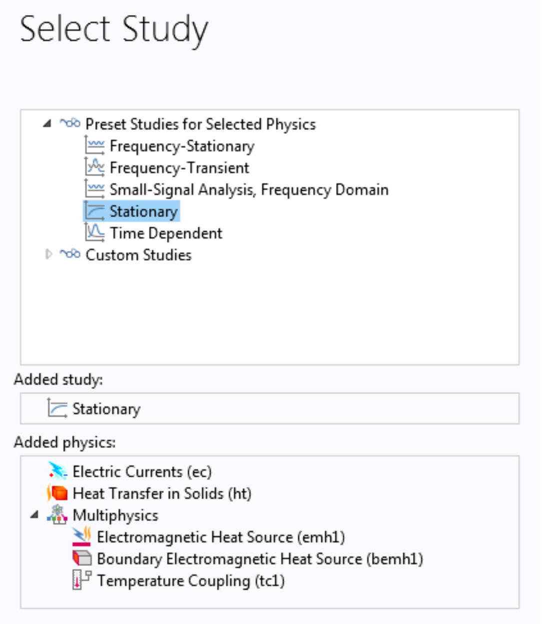
To remove an interface already selected, highlight it in the list and click **Remove** under the table.

SELECT STUDY

On the **Select Study** page, click to select the type of study to perform. The available options depend on the set of interfaces included in the model. Some study types are applicable to all physics interfaces for which you choose to solve, while others are not, but all are in some way available. You can select the study type from one of the following branches (see Figure 2-12):

- **Preset Studies**—Studies suggested by a single physics interface if only one has been chosen.
- **Preset Studies for Selected Physics**—Studies applicable to all physics interfaces that you have chosen to solve for.

- **Custom Studies**—In some study types, not all physics solved for can generate suitable equations. The custom studies branch accounts for those study types:
  - **Preset Studies for Some Physics**—The study types recognized by some, but not all, of the physics being solved for.
  - **Empty Study**
  - Other studies—Any fundamental study types (**Stationary**, **Time Dependent**, **Eigenfrequency**, **Eigenvalue** and **Frequency Domain**) which are not applicable to any of the physics being solved for. There is also an empty study type.



Select Study

- Preset Studies for Selected Physics
  - Frequency-Stationary
  - Frequency-Transient
  - Small-Signal Analysis, Frequency Domain
  - Stationary
  - Time Dependent
- Custom Studies

Added study:

Stationary

Added physics:

- Electric Currents (ec)
- Heat Transfer in Solids (ht)
- Multiphysics
  - Electromagnetic Heat Source (emh1)
  - Boundary Electromagnetic Heat Source (bemh1)
  - Temperature Coupling (tc1)

Figure 2-12: The Select Study page in the Model Wizard.

Once a study is highlighted, information about it displays to the right of the window and it is included under **Added study**. there are several options to continue. Click:

- **Add** to add the physics interface to the **Added physics** list, or right-click and select **Add physics**. Add as many interfaces as you want. You can use the **Review Physics** page to edit the Dependent Variables as required.
- **Physics** to go back to the **Select Physics** page.
- **Done** to exit the Model Wizard.
- **Cancel** to return to the COMSOL Desktop.
- **Help** to open the context-based **Help** window.

After clicking **Done**, the **Model Builder** window displays a model tree with a set of default nodes in the **Component** branches—**Definitions**, **Geometry**, **Materials**, **Mesh**, and nodes based on the physics interfaces selected (see

Figure 2-13). The **Component** nodes and branches form the sequence of operations that define the model.

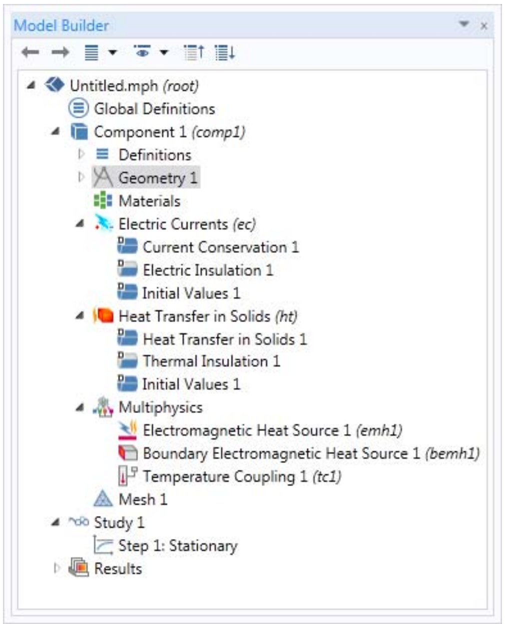


Figure 2-13: After clicking Finish, a 3D Component with a Solid Mechanics interface and Eigenfrequency study is added to the Model Builder.

Also add a study from the **Model Builder** and The Add Study Window:

- By right-clicking the **Root** node and selecting **Add Study**.
- On the **Home** ribbon (Windows users) or the **Model Toolbar** (Mac and Linux users), click the **Add Study** button.

- Building a COMSOL Model
- The Component Node
- The Add Physics Window
- The Add Study Window
- The Model Builder

Toolbars and Keyboard Shortcuts

The toolbars and context menus in COMSOL Multiphysics are based on the stage of modeling. This section is a single resource for each of the ribbon and contextual toolbars available on the COMSOL Desktop. There are also several Keyboard Shortcuts that are useful for navigating during the modeling process. The following sections have a table where there are links to more information about the available items on the ribbon toolbar or contextual toolbar.

- Home Ribbon (Windows) and Model Toolbar (Mac/Linux)
- Definitions Toolbar
- Geometry Toolbar
- Physics Toolbar
- Mesh Toolbar
- Study Toolbar
- Results Toolbar
- Plot Group Contextual Toolbar
- Work Plane Modal Toolbar

About Changes to the Ribbon Display (Windows Users)

Home Ribbon (Windows) and Model Toolbar (Mac/Linux)

The **Home** ribbon toolbar (Windows) and the **Model Toolbar** (Mac and Linux) contains many of the common features and actions required to build and analyze a model.

TABLE 2-4: THE HOME RIBBON (WINDOWS) OR MODEL TOOLBAR (MAC AND LINUX)

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Model			
—	Component	Once 3D, 2D, 2D axisymmetric, 1D, and 1D axisymmetric Components are available or added, these are listed here and you can click to take you to the node in the Model Builder.	The Component Node
◈	Add Component	3D, 2D, 2D Axisymmetric, 1D, 1D Axisymmetric, 0D	The Component Node
Definitions			
Pi	Parameters	Add globally available Parameters to your model.	About Global and Local Definitions and Parameters
a=	Variables	Choose from Global Variables and Local Variables.	About Global and Local Definitions, Predefined and Built-In Variables
f(x)	Functions	Choose from a list of all available Functions	Functions

TABLE 2-4: THE HOME RIBBON (WINDOWS) OR MODEL TOOLBAR (MAC AND LINUX)

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Geometry			
▦	Build All	Build all features in the current geometry.	Editing and Building Geometry Nodes The Geometry Node
⇐	Import	Import the geometry from a COMSOL Multiphysics file or CAD file.	Import
⇄	LiveLink	Choose from a list of available LiveLink products.	Overview of Geometry Modeling Concepts
Materials			
▦	Browse Materials	Open the Material Browser where you can access and edit material libraries.	The Material Browser Window and The Add Material Window
▦	New Material	Add an empty material to your model.	Materials and Creating a New Material Library
▦	Add Material	Open the Add Material window to add materials to components or selections.	The Add Material Window
Physics			
⚛	Select Physics	For a blank model this is available.	Selecting Physics
—	Physics	When interfaces are added, these are listed here and you can click to take you to the node in the Model Builder.	Physics Guide
⚛	Add Physics	Open the Add Physics window to add physics to the current component.	The Add Physics Window
Mesh			
▦	Build Mesh	Build the current mesh.	Adding, Editing, and Building Meshing Sequences
△	Select Mesh	Available for a blank model.	Mesh Elements for 1D, 2D, and 3D Geometries
△	Mesh	When meshes are added, these are listed here and you can click to take you to the node in the Model Builder.	Creating a Mesh for Analysis
Study			
=	Compute	Compute the selected study.	Computing a Solution
∞	Select Study	Available for a blank model.	Open a New Window to Begin Modeling and The Add Study Window
∞	Study	When studies are added, these are listed here and you can click to take you to the node in the Model Builder.	Introduction to Solvers and Studies

TABLE 2-4: THE HOME RIBBON (WINDOWS) OR MODEL TOOLBAR (MAC AND LINUX)

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Add Study	Open the Add Study window to add a study to the current Component.	The Add Study Window
Results			
1D 2D 3D	Select Plot Group	Available for a blank model.	About the Plot Groups
	Add Plot Group	3D Plot Group, 2D Plot Group, 1D Plot Group, Polar Plot Group.	About the Plot Groups
Table 2 0-7	(3D, 2D, 1D or Polar) Plot Group	Once results are available or added, these are listed here and you can click to take you to the node in the Model Builder. It also opens a new Plot Group contextual toolbar for the plot group.	Plot Groups and Plots

For cross-platform users, the options listed in Table 2-5 are available from other toolbars and menus. See Cross Platform (Mac and Linux) Toolbars and Menus.

TABLE 2-5: AVAILABLE ONLY ON THE HOME RIBBON (WINDOWS)

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Model Libraries	Open the Model Libraries window.	The Model Libraries Window
	More Windows	Add Physics	Creating a New Model
		Add Study	The Model Wizard
		Add Material	The Material Browser Window
		Material Browser	The Material Browser Window
		Selection List	The Selection List Window
		Messages	The Messages Window
		Table	The Table Window and Tables Node
		External Process	The External Process Window
		Physics Builder Manager	See the *Physics Builder Manual*.
Layout			
	Reset Desktop	Reset the COMSOL Desktop to its default settings.	Customizing the Desktop Layout
	Desktop Layout	Choose Widescreen Layout or Regular Screen Layout,	Customizing the Desktop Layout
	Model Builder Node Label	Choose from Show Name Only, Show Name and Identifier, Show Name and Tag, Show Type and Identifier, or Show Type and Tag.	Viewing Node Names, Identifiers, Types, and Tags

Definitions Toolbar

The **Definitions** ribbon toolbar (Windows) and the **Definitions** contextual toolbar (Mac and Linux) contain many of the common features and actions required to work with features found under the Definitions node in the Model Builder.

For step-by-step instructions and general documentation descriptions, this is the **Definitions** toolbar.

TABLE 2-6: THE DEFINITIONS TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Variables			
a=	Local Variables	Add a Variables node under Definitions.	About Global and Local Definitions, Predefined and Built-In Variables
Functions			
f(x)	Analytic	Add an Analytic node under Definitions.	Analytic
	Interpolation	Add an Interpolation node under Definitions.	Interpolation
	Piecewise	Add an Piecewise node under Definitions.	Piecewise
f(x)	More Functions	All available Functions (except Analytic, Interpolation, and Piecewise).	Functions
Selections			
	Explicit	Add an Explicit node under Definitions.	Named Selections and Explicit
	Complement	Add a Complement node under Definitions.	Union, Intersection, Difference, and Complement
	Adjacent	Add an Adjacent node under Definitions.	Adjacent
	Ball	Add a Ball node under Definitions.	Ball
	Box	Add a Box node under Definitions.	Box
	Cylinder	Add a Cylinder node under Definitions.	Cylinder
	Union	Add a Union node under Definitions.	Union, Intersection, Difference, and Complement
	Intersection	Add an Intersection node under Definitions.	
	Difference	Add a Difference node under Definitions.	

TABLE 2-6: THE DEFINITIONS TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Probes			
	Update Probes	Update all probes.	Probes
	Probes	Select an option from the list to add a node under Definitions Table 5-16.	
Coupling			
	Component Couplings	Select an option from the list to add a node under Definitions.	Component Couplings
	Pairs	Select an option from the list to add a node under Definitions.	About Identity and Contact Pairs
Coordinate Systems			
	Coordinate Systems	Select an option from the list to add a node under Definitions Table 5-15.	Coordinate Systems
	Perfectly Matched Layer	Add a Perfectly Matched Layers node under Definitions.	Perfectly Matched Layer
	Infinite Element Domains	Add an Infinite Element Domains node under Definitions.	Infinite Element Domain
View			
	View	Add a View node to the current Component to control the view and lighting in the Graphics window.	View (2D), View (3D), Axis
	Reset to Default	Restores the default settings for the current View node.	User-Defined Views

Geometry Toolbar

Once a geometry is added to the model, the **Geometry** ribbon toolbar (Windows) and the **Geometry** contextual toolbar (Mac and Linux) contains many of the common features and actions required to create and build a geometry.

For cross-platform users, some options listed in Table 2-7 are available from other toolbars and menus. Table 2-8 lists the geometry drawing tools available on the toolbars in 1D and 2D, as well as on the Work Plane toolbar for 3D models. See Cross Platform (Mac and Linux) Toolbars and Menus.

For step-by-step instructions and general documentation descriptions, this is the **Geometry** toolbar.

TABLE 2-7: THE GEOMETRY TOOLBARS

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS
Build		
	Build All[1]	Build all features in the current geometry. See Editing and Building Geometry Nodes.
Import/Export		
	Import[1]	Import the geometry from a COMSOL Multiphysics file or CAD file.
	Insert Sequence	Insert a geometry sequence from a COMSOL Multiphysics file into the current geometry. See Insert a Sequence.
	Export	Export the current geometry as a COMSOL binary or text file or to a CAD file format. See Exporting a Geometry.
Primitives (in 3D) See Geometric Primitives.		
	Block	Add a Block to the current geometry.
	Cone	Add a Cone to the current geometry.
	Cylinder	Add a Cylinder to the current geometry.
	Sphere	Add a Sphere to the current geometry.
	Torus	Add a Torus to the current geometry.
	Helix	Add a Helix to the current geometry.
	More Primitives	Add all other primitives as in Table 7-1.
Primitives (in 2D)		
	More Primitives	Interpolation Curve, Parametric Curve, and Point. Also see Geometric Primitives and The Geometry Toolbar—Drawing Tools (Table 2-8).
Primitives (in 1D)		
See The Geometry Toolbar—Drawing Tools (Table 2-8).		
Work Plane		
	Work Plane (1, 2, 3...)	Once Work Planes are available or added, these are listed here and you can click to take you to the associated Plane Geometry node in the Model Builder. It also opens a new Work Plane contextual toolbar for the Work Plane. See Using Work Planes, Work Plane Modal Toolbar and Table 2-8.
	Work Plane	Add a Work Plane to the current geometry. See Using Work Planes, Work Plane Modal Toolbar and Table 2-8.
Operations		
	Extrude	Extrude planar faces of geometry objects or objects from a work plane to create 3D geometry objects.
	Revolve	Revolve planar faces of geometry objects or objects from a work plane about an axis to create 3D geometry objects.
	Sweep	Sweep faces along a spine curve to create a solid object.

TABLE 2-7: THE GEOMETRY TOOLBARS

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS
	Union	Create a geometry object as the set union of other geometry objects.
	Intersection	Create a geometry object as the set intersection of other geometry objects.
	Difference	Create a geometry object as the set difference between two other geometry objects (or sets of geometry objects).
	Compose	Compose a geometry object from existing geometry objects using a set formula containing Boolean operators.
	Partition	Partition geometry objects into parts using other geometry objects as tool objects for the partition. Alternatively, in 3D, use a work plane to partition the geometry objects.
	Array	Create an array (linear or rectangular in 2D, linear or three-dimensional in 3D) of identical geometry objects.
	Copy	Copy geometry objects, creating a copy with an optional displacement from the original geometry objects.
	Mirror	Mirror geometry objects in a plane (in 3D) or line (in 2D).
	Move	Move geometry objects.
	Rotate	Rotate geometry objects about an axis of rotation.
	Scale[2]	Scale geometry objects using isotropic or anisotropic scaling.
	Conversions[2]	See Table 7-5 for a list of these features. Also see Conversion Operations.
	Chamfer	Chamfer corners in 2D geometry objects.
	Fillet	Fillet corners in 2D geometry objects.
	Tangent	Create a line segment tangent to a specified edge.
	Delete[2]	Delete the selected geometry objects. See Clearing Sequences and Disabling, Enabling, and Deleting Nodes.
	Defeaturing and Repair[2,3]	Choose from Cap Faces, Delete Fillets, Delete Short Edges, Delete Sliver Faces, Delete Small Faces, Delete Spikes, Delete Faces, Detach Faces, Knit to Solid, and Repair.
	Virtual Operations[2]	Virtual geometry operations and mesh control operations. See Virtual Geometry Operations and Virtual Geometry and Mesh Control Operations.
Other		
	Programming[2]	If + End If, Subsequence Call. See Using Geometry Subsequences and If, Else If, Else, End If.
	Selections[2]	Create named selections of geometry objects or geometric entities in geometry objects. See Creating Named Selections in the Geometry Sequence.

TABLE 2-7: THE GEOMETRY TOOLBARS

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS
	Measure[2]	Measure the volume, area, perimeter, or other geometric properties of the selected geometric entities or objects. See Measuring Geometry Objects.

[1] For cross-platform users, this option is available from a different toolbar or menu. See Cross Platform (Mac and Linux) Toolbars and Menus.

[2] For cross platform users (Mac and Linux), these buttons may only be visible if the Tools>Toolbar Display Mode is set to Compact or you are on a widescreen monitor. It also depends if the button is available on the Work Plane toolbar, in which case it may be visible.

[3] These features are available for the LiveLink and CAD products.

GEOMETRY DRAWING TOOLBAR BUTTONS

In 1D and 2D there are buttons for drawing Geometric Primitives by using the mouse. In 3D, the buttons are available to create primitives, but you cannot draw these using the mouse unless you are using a Work Plane.

TABLE 2-8: THE GEOMETRY TOOLBAR—DRAWING TOOLS

BUTTON OR MENU	NAME	DESCRIPTION OR ACTION
Draw Settings		
	Snap Coordinates	Snap to the grid when drawing a geometry object in the Graphics window. By default, the mouse pointer snaps to the grid points and geometry vertices (for example, the corners of a rectangle). To disable snapping, click the Snap Coordinates button.
	Draw Solid	Create a solid (instead of a curve) when drawing a geometry object in the Graphics window. Available for 2D and 2D axisymmetric Components. The Draw Solid button is used to toggle between drawing solid objects or outlines (curves) and can be used in combination with all the draw buttons on the toolbar (for example, to draw squares, circles, and rectangles). It is selected by default to draw solid objects. When used to draw a polygon in combination with the Draw Line, Draw Quadratic, and Draw Cubic operations, the solid object (or outline) displays once the object is closed or the points joined.
Primitives (in 2D)		
	Draw Line	Draw line segments, quadratic curve segments, and cubic curve segments of a Bézier Polygon in the Graphics window. To draw a polygon consisting of line segments or Bézier curves first click one of these buttons. Then click the control points of the segments in the Graphics window. Click one point for each linear segment, two points for each quadratic segment, and three points for each cubic segment. If you want to switch segment type, click one of the buttons and then click some more control points. Close the polygon by right-clicking anywhere in the Graphics window. Then, a solid Bézier polygon appears, and a corresponding Bézier Polygon node appears in the geometry sequence. If you want to modify the polygon (for instance, change from solid to curve) you can edit the Bézier Polygon node by clicking it to display its settings window.
	Draw Quadratic	
	Draw Cubic	

TABLE 2-8: THE GEOMETRY TOOLBAR—DRAWING TOOLS

BUTTON OR MENU	NAME	DESCRIPTION OR ACTION
□	Rectangle	To draw a rectangle select one of these buttons from the Rectangle □ menu. Then, click one corner (or the center) of the rectangle in the Graphics window. Drag the mouse to the desired position of a corner. When the mouse button is released, a solid rectangle appears and a Rectangle node is added to the geometry sequence.
▣	Rectangle (Center)	
□	Square	To draw a square select one of these buttons from the Rectangle □ menu. Then, click one corner (or the center) of the square in the Graphics window. Drag the mouse to the desired position of a corner. When the mouse button is released, a solid square appears and a Square node is added to the geometry sequence.
▣	Square (Center)	
◉	Circle	To draw a circle select one of these buttons from the Circle ◉ menu. Then, click the circle's corner (or one corner of the circle's bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the circle's bounding box. When the mouse button is released, a solid circle appears and an Circle node is added to the geometry sequence.
○	Circle (Corner)	
⬬	Ellipse	To draw an ellipse select one of these buttons from the Circle ◉ menu. Then, click the ellipse's corner (or one corner of the ellipse's bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the ellipse's bounding box. When the mouse button is released, a solid ellipse appears and an Ellipse node is added to the geometry sequence.
⬭	Ellipse (Corner)	
Primitives (in 1D)		
╱	Interval	Draw an Interval (line) in the Graphics window. For 1D models, first click the Interval button, then click the start and end points in the Graphics window.
▪	Point (also available in 2D and 3D)	Add a point to the Graphics window. Use this to draw a single point. First click the Point button, then click in the Graphics window (in 1D and 2D), or specify the point location in its settings window. In 2D and 3D, this button is available on the More Primitives menu.

Work Plane Modal Toolbar

The **Work Plane** modal toolbar is available after clicking Plane Geometry Node under **Geometry>Work Plane** in the Model Builder.

For step-by-step instructions and general documentation descriptions, this is the **Work Plane** modal toolbar.

The Work Plane toolbar is similar to the 2D Geometry Toolbar except that the **Virtual Operations** menu is not available and there is a **Close** button.

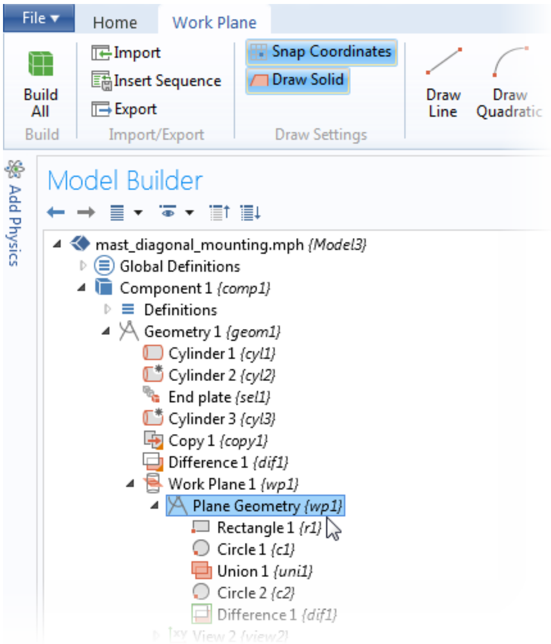


Figure 2-14: Click the Plane Geometry node to open the Work Plane toolbar.

Physics Toolbar

Once physics interfaces are added to the model, the **Physics** ribbon toolbar (Windows) and the **Physics** contextual toolbar (Mac and Linux) contains many of the common features and actions required to add physics and features to the Model Builder.

For step-by-step instructions and general documentation descriptions, this is the **Physics** toolbar.

TABLE 2-9: THE PHYSICS TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Physics			
Various	Select Physics	Any physics interfaces added to the selected Component are listed. Click to go to the node in the Model Builder and open the settings window. See Physics Guide.	The Physics Interfaces
⚛	Add Physics	Open the Add Physics window to add physics to the current component.	The Add Physics Window
Geometric Entity			

TABLE 2-9: THE PHYSICS TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Domains	See Table 3-3 for a list of all the icons by space dimension. Available physics features for the physics interface are listed. To add subfeatures, however, you need to right-click the parent node. For example, to add a Destination Selection subnode, right-click Periodic Condition.	About Geometric Entities About Selecting Geometric Entities
	Boundaries		
	Pairs		
	Edges		
	Points		
	Global		

Mesh Toolbar

Once a mesh is added to the model, the **Mesh** ribbon toolbar (Windows) and the **Mesh** contextual toolbar (Mac and Linux) contains many of the common features and actions required to work with meshes.

TABLE 2-10: THE MESH TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Build			
	Build Mesh[1]	Build the current mesh.	Adding, Editing, and Building Meshing Sequences
	Mesh (1, 2, 3...)	Lists the meshes available in the model. Click and you go to the node in the Model Builder	Creating a Mesh for Analysis
	Add Mesh	Add a new mesh to the current model component.	Adding, Editing, and Building Meshing Sequences
Physics-Controlled			
	Edit	Edit the physics-controlled sequence.	Physics-Controlled Mesh
	Reset	Reset the sequence to the physics-controlled settings.	
Mesh Size			
—	Mesh Size	See Table 8-1 for a list of options.	The Mesh Toolbar (Predefined Mesh Element Sizes) Mesh Element Quality and Size
Generators			
—	Inherit Size	See Table 8-1 for a list of options.	Predefined Mesh Element Sizes and Mesh Element Quality and Size
	Free Tetrahedral	Generate unstructured tetrahedral mesh.	Meshing Operations and Attributes

TABLE 2-10: THE MESH TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Swept	Generate swept mesh.	
	Boundary	Free Triangular, Free Quad, Mapped, and Edge.	
	Boundary Layers	Generate boundary layer mesh.	
Operations			
	Modify	Elements: Convert and Refine Size: Distribution and Corner Refinement	Meshing Operations and Attributes and Mesh Attributes
	Copy	Copy Domain, Copy Edge, and Copy Face	Meshing Operations and Attributes
Import			
	Import	Import mesh.	Importing Meshes
	Partition	Ball, Box, Cylinder, Logical Expression, and Create Vertex.	Partition the Geometry into Simple Domains and Partition
	Combine	Delete Entities and Join Entities.	
Evaluate			
	Measure	Measure the volume, area, perimeter, or other geometric properties of the selected geometry objects.	Measuring Geometry Objects
	Statistics	Write information about number and quality of elements to the Messages window.	The Mesh Statistics Window
	Plot	Add a mesh plot.	Mesh (Plot)
Clear			
	Clear Mesh	Clear the mesh.	Clearing Sequences and Disabling, Enabling, and Deleting Nodes
	Clear All Meshes	Clear all meshes in the model.	

[1] For cross-platform users, this option is available from a different toolbar or menu. See Cross Platform (Mac and Linux) Toolbars and Menus.

Study Toolbar

The **Study** ribbon toolbar (Windows) and the **Study** contextual toolbar (Mac and Linux) contains many of the common features and actions required to work with studies and solvers.

For step-by-step instructions and general documentation descriptions, this is the **Study** toolbar.

TABLE 2-11: THE STUDY TOOLBAR

BUTTON OR MENU	NAME	OPTIONS DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Study			
	Compute[1]	Compute the selected study.	Computing a Solution
	Select Study[1]	See a list of all available studies in Table 19-2 and Table 19-3.	Study Types
	Add Study[1]	Open the Add Study window to add a study to the current model component.	The Add Study Window
	Continue	Continue the computation of a solver sequence from the last computed feature.	The Progress Window
	Update Solution	Update the solution using the current values of parameters and user-defined variables.	Updating a Solution
u t=0	Get Initial Value	Generate a solution using the initial values of the dependent variables (without solving for the dependent variables).	Computing the Initial Values
Solver			
	Show Default Solver	Display the nodes in the solver sequences that are created by default.	Show Default Solver
Study Step			
	Study Steps	See a list of all available studies in Table 19-2 and Table 19-3.	The Add Study Window and The Relationship Between Study Steps and Solver Configurations
	Parametric Sweep	Choose Parametric Sweep or Optimization, for example.	Study Extension Steps and Advanced Study Extension Steps
Operations			
	Create Solution Copy	Make a copy of a solution.	Solution Operation Nodes and Solvers
Evaluate			

TABLE 2-11: THE STUDY TOOLBAR

BUTTON OR MENU	NAME	OPTIONS DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Statistics	Display statistics for the study, including the number of degrees of freedom (DOFs).	The Statistics Page
Clear			
	Clear Solutions	Clear the solutions in the current solver sequence.	Clearing Sequences and Disabling, Enabling, and Deleting Nodes
	Clear All Solutions	Clear all solutions.	

[1] For cross-platform users, this option is available from a different toolbar or menu. See Cross Platform (Mac and Linux) Toolbars and Menus.

Results Toolbar

The **Results** ribbon toolbar (Windows) and the **Results** contextual toolbar (Mac and Linux) contains many of the common features and actions required to work with studies and solvers.

For step-by-step instructions and general documentation descriptions, this is the **Results** toolbar.

TABLE 2-12: THE RESULTS TOOLBAR

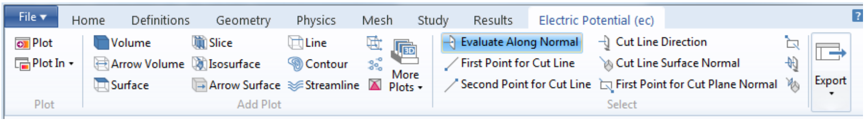
BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
Plot Group			
	Plot	Plot the selected plot group.	Plot Groups and Plots. Also see Plot Group Contextual Toolbar.
—	Select Plot Group	Choose from a list of plots included in the model.	About the Plot Groups. Also see Plot Group Contextual Toolbar.
or	3D Plot Group	Create a new 3D Plot Group for 3D plots such as volume and slice plots.	
or	2D Plot Group	Create a new 2D Plot Group for 2D plots such as surface and contour plots.	
or	1D Plot Group	Create a new 1D Plot Group for 1D graph plots.	
	Polar Plot Group	Create a new Polar Plot Group for graph plots in a polar coordinate system.	
Data Set			
	Cut Plane	Create a Cut Plane data set for data on a plane in 3D.	Cut Plane
	Cut Line 3D	Create a 3D Cut Line data set for data along a line in 3D.	Cut Line 2D and Cut Line 3D

TABLE 2-12: THE RESULTS TOOLBAR

BUTTON OR MENU	NAME	DESCRIPTION OR OPTIONS	LINK TO MORE INFORMATION
	Cut Point 3D	Create a 3D Cut Point data set for data at a point in 3D.	Cut Point 1D, Cut Point 2D, and Cut Point 3D
	Cut Line 2D	Create a 2D Cut Line data set for data along a line in 2D.	Cut Line 2D and Cut Line 3D
	Cut Point 2D	Create a 2D Cut Point data set for data at a point in 2D.	Cut Point 1D, Cut Point 2D, and Cut Point 3D
	More Data Sets	See Table 20-6 for a list of all data sets.	Data Sets
Derived Values			
=	Evaluate All	Evaluate all derived values.	Table Window toolbar and Menu Options
	Clear and Evaluate All	Clear all current table entries and then evaluate all derived values.	Table Window toolbar and Menu Options
8.85 e-12	Point Evaluation	Add a Point Evaluation node to evaluate some expressions or variables at points.	Point Evaluation
8.5	Global Evaluation	Add a Global Evaluation node to evaluate some global expressions or variables.	Global Evaluation
8.85 e-12	More Derived Values	See Table 20-10 for a list of all derived value types.	Derived Values and Tables
Export			
	Data	Data, Plot, Mesh (Export), Table	Exporting Data and Images
	Image	1D Image, 2D Image, or 3D Image	
	Animation	Create a movie to animate a solutions (as an animated GIF, Flash, or AVI movie file).	
	Player	Create a Player for generating animation directly in the COMSOL Desktop Graphics window.	
Report			
	Report	Choose a Brief Report, Intermediate Report, Complete Report, Custom Report, or Documentation.	Report Types

Plot Group Contextual Toolbar

The plot group contextual toolbar is available after clicking a specific plot group in the Model Builder. The available tools are based on the model and the type of plot.



For step-by-step instructions and general documentation descriptions, this is generally referred to as the **Plot Group** toolbar, where the name of the toolbar name changes based on the plot group (3D Plot Group, 2D Plot Group, 1D Plot Group, or Polar Plot Group). If the plot group is renamed, the toolbar name also changes to match the new name, as in the example above where the **3D Plot Group** toolbar was renamed **Electric Potential**.

TABLE 2-13: THE PLOT GROUP CONTEXTUAL TOOLBAR

BUTTON OR MENU	NAME	LINK TO MORE INFORMATION
Plot		
	Plot	Plot Groups and Plots
	Plot In New Window	The Plot Windows
Add Plot		
Various	See Table 20-7 for the available plots by Plot Group (3D, 2D, 1D, or Polar Plot Groups).	About the Plot Groups
Select (available for creating cross-section plots)		
Various	See Table 20-8 for a list of the available buttons.	Creating Cross-Section Plots and Combining Plots and Plotting and Cross-Section Interactive Toolbar
Export		
Various	See Table 20-11 for a list of export options.	Exporting Data and Images

Keyboard Shortcuts

The following table summarizes the available keyboard shortcuts on Windows and Linux and on Macintosh:

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
F1	F1	Display help for the selected node or window
Ctrl+F1	Command+F1	Open the COMSOL Documentation front page in an external Help window
F2	F2	Rename the selected node, file, or folder
F3	F3	Disable selected nodes

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
F4	F4	Enable selected nodes
F5	F5	Update solution with respect to new definitions without re-solving the model
F7	F7	Build the selected node in the geometry and mesh branches, or compute to the selected node in the solver sequence
F8	F8	Build the geometry, build the mesh, compute entire solver sequence, update results data, or update the plot
Del	Del	Delete selected nodes
Left arrow (Windows); Shift + left arrow (Linux)	Left arrow	Collapse a branch in the Model Builder
Right arrow (Windows); Shift + right arrow (Linux)	Right arrow	Expand a branch in the Model Builder
Up arrow	Up arrow	Move to the node above in the Model Builder
Down arrow	Down arrow	Move to the node below in the Model Builder
Alt+left arrow	Ctrl+left arrow	Move to the previously selected node in the Model Builder
Alt+right arrow	Ctrl+right arrow	Move to the next selected node in the Model Builder
Ctrl+A	Command+A	Select all domains, boundaries, edges, or points; select all cells in a table
Ctrl+D	Command+D	Clear the selection of all domains, boundaries, edges, or points
Ctrl+C	Command+C	Copy text in fields
Ctrl+N	Command+N	Create a new model
Ctrl+O	Command+O	Open a model file
Ctrl+P	Command+P	Print the contents of the plot window
Ctrl+S	Command+S	Save a model file
Ctrl+V	Command+V	Paste copied text
Ctrl+Z	Command+Z	Undo the last operation
Ctrl+Y	Ctrl+Shift+Z	Redo the last undone operation
Ctrl+up arrow	Command+up arrow	Move a definitions node, geometry node, physics node (except default nodes), material node, mesh node, study step node, or results node up one step
Ctrl+down arrow	Command+down arrow	Move a definitions node, geometry node, physics node (except default nodes), material node, mesh node, study step node, or results node down one step

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
Ctrl+Tab	Ctrl+Tab	Switch focus to the next window on the desktop
Ctrl+Shift+Tab	Ctrl+Shift+Tab	Switch focus to the previous window on the desktop
Ctrl+Alt+left arrow	Command+Alt+left arrow	Switch focus to the Model Builder window
Ctrl+Alt+right arrow	Command+Alt+right arrow	Switch focus to the settings window
Ctrl+Alt+up arrow	Command+Alt+up arrow	Switch focus to the previous section in the settings window
Ctrl+Alt+down arrow	Command+Alt+down arrow	Switch focus to the next section in the settings window
Shift+F10 or (Windows only) Menu key	Ctrl+F10	Open context menu
Ctrl+Space	Ctrl+Space	Open list of predefined quantities for insertion in Expression fields for plotting and results evaluation.

3

Building a COMSOL Model

This chapter explains a range of methods and topics used when building models in COMSOL Multiphysics: From working with the Model Builder and fundamental concepts for building a model to the use of units. For examples of how to build a complete model step-by-step, see the model library for COMSOL Multiphysics® and the add-on modules and the *Introduction to COMSOL Multiphysics* book.

Building Models in the Model Builder

The power of COMSOL Multiphysics is the ease of working with all the components required to build a model in The Model Builder. The sections About the Sequence of Operations, The Component Node, Branches and Subbranches in the Tree Structure, The Node Settings Windows, and Opening Context Menus and Adding Nodes further introduce you to key concepts about navigating in the Model Builder, the structure of the tree, and how to add features (nodes) as you build your model.

The physics feature nodes that are added to physics interfaces are flexible and several sections describe the ways to identify changes, status updates, and other ways to work with these nodes: The Physics Nodes, Physics Default Nodes, Physics Nodes by Space Dimension, Physics Node Context Menu Layout, Physics Exclusive and Contributing Node Types, Physics Node Status, Dynamic Nodes in the Model Builder, and Errors and Warnings.

- The Root Window
- Creating a New Model
- The COMSOL Desktop

The Model Builder

The modeling procedure is controlled through the **Model Builder** window, which is essentially a *model tree* with all the functionality and operations for building and solving models and displaying the results. These are introduced to your modeling procedure by adding a *branch*, such as the Geometry branch. Branches can have further *nodes* (or *subbranches*) that relate to their parent node. It is all About the Sequence of Operations. See Figure 3-2 for an example.

A node has its own properties and *settings* that are characteristic to it. Branches and subbranches can also contain properties and settings. See Branches and Subbranches in the Tree Structure and The Node Settings Windows for examples.

The Model Builder has many types of nodes to help you create models and visualize the model structure, for example, the Component node is categorized by space dimension, and nodes are dynamic, which helps you identify nodes that change status. See Component Nodes by Space Dimension, Physics Default Nodes, and Dynamic Nodes in the Model Builder for more information.

Also learn about the context menu available when you right-click a node in the Model Builder (Opening Context Menus and Adding Nodes). In the next section (Working with Nodes in the Model Builder), there is also information about Going to the Source Node, Copying, Pasting, and Duplicating Nodes, Undoing and Redoing Operations, and Clearing Sequences and Disabling, Enabling, and Deleting Nodes.

- The Root Window
- Creating a New Model
- Basic Navigation
- The COMSOL Desktop

About the Sequence of Operations

COMSOL operates through *sequencing* and evaluates most of the branch nodes in the Model Builder from the top-down as a *sequence of operations*. By adding nodes to a branch in the Model Builder in a certain order, you set up such sequences of operations, which makes it possible to, for example, parameterize a model and rerun the simulation. COMSOL then re-evaluates each sequence, automatically updating the geometry, mesh, physics, and solution. A solver sequence, for example, could define your model with one solver and then, using the returned solution, solve it with an alternative solver.

For most sequences, you run the sequence by right-clicking the top node of the branch and selecting **Build All** (geometry and mesh), **Compute** (studies), or **Plot** (plot groups), or by pressing F8.

Some nodes under a physics branch can override other nodes higher up in the sequence. How COMSOL treats those nodes depends on whether they are contributing or exclusive nodes (see Physics Exclusive and Contributing Node Types).

The sequence of operations means that the order of the nodes in the tree is important. In the following branches of the model tree, the node order makes a difference, and you can move nodes up and down to change the sequence of operations for these nodes—Geometry, Material, Physics, Mesh, and Solver.

Also, the order can have some importance in the plot groups in the Results branch and also for the Perfectly Matched Layers and Infinite Elements nodes in the Definitions branch (those nodes are available with some of the add-on modules).

- Physics Node Status
- Physics Exclusive and Contributing Node Types
- Creating a Geometry for Analysis and Working with Geometry Sequences
- Moving Nodes in the Model Builder

The Component Node

Previously this feature was named *Model*.

A model component is a fundamental part of the model and contains a geometry with its associated physics, mesh, and variables and other definitions that are local to that component. The **Component** node defines the namespace for each part of a model that is defined in a model component. A model can have several **Component** nodes. For example, if you are setting up a system model using both a 2D simplification—represented in one 2D **Component** branch—and a full 3D description in another **Component**, these can both be added to the Model Builder to represent different aspects or parts of the model. You can couple variables between different components in a model using coupling operators.

To **Add Physics**, **Add Extra Dimension**, and **Add Mesh** to the Component, from the **Home** ribbon (Windows users), **Model Toolbar** (Mac and Linux users), or for any operating system, right-click the **Component** node. See Adding Extra Dimensions to a Component, The Add Physics Window, and Creating a Mesh for Analysis for more information.

To enable the **Extra Dimensions** option, go to The Preferences Dialog Box. Click **Model Builder**, select the **Enable technology preview functionality** check box, and then click **OK**.

The **Component** node icon also indicates the space dimension:.

TABLE 3-1: SPACE DIMENSION ICONS IN THE MODEL BUILDER

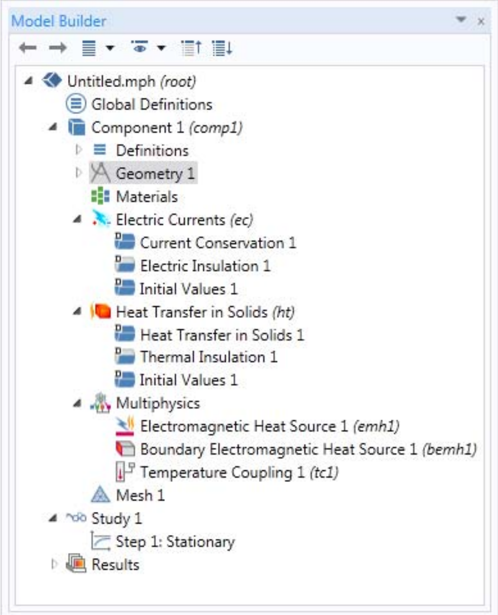
ICON	SPACE DIMENSION
■	3D
⊥	2D axisymmetric
Ω	2D
⊥	1D axisymmetric
—	1D
·	0D (space-independent models for chemical reactions and other ODEs/DAEs)

Adding a Component to a Model

You can create models with multiple geometries by adding one or more Component nodes to the Model Builder.

To add a **Component** node or nodes:

- Right-click the **Root** node (the top most node) in the **Model Builder** and select **Add Component** (see The Root Window).
- In The Model Wizard on the **Select Space Dimension** page, select **3D**, **2D axisymmetric**, **2D**, **1D axisymmetric**, or **1D**. Continue defining the model as in Creating a New Model.



Model Builder

- Untitled.mph *(root)*
  - Global Definitions
  - Component 1 *(comp1)*
    - Definitions
    - Geometry 1
    - Materials
    - Electric Currents *(ec)*
      - Current Conservation 1
      - Electric Insulation 1
      - Initial Values 1
    - Heat Transfer in Solids *(ht)*
      - Heat Transfer in Solids 1
      - Thermal Insulation 1
      - Initial Values 1
    - Multiphysics
      - Electromagnetic Heat Source 1 *(emh1)*
      - Boundary Electromagnetic Heat Source 1 *(bemh1)*
      - Temperature Coupling 1 *(tc1)*
    - Mesh 1
  - Study 1
    - Step 1: Stationary
  - Results

Figure 3-1: An example of the Model Builder default nodes for the Electric Currents and Heat Transfer in Solids interfaces.

These default nodes are normally added under a **Component** node:

- **Definitions**: Contains user-defined variables, selections, views, pairs, functions, probes, component couplings, and coordinate systems, which are defined locally for the model. See About Global and Local Definitions for information about using these local **Definitions** (≡) and **Global Definitions** (≡). Use **Global Definitions** to define **Parameters**, **Variables**, **Functions**, and **Groups** with a global scope—that is, not specific to one **Component** node.
- **Geometry** ( ): Contains the sequence of geometric objects and operations (or imported CAD data) that defines the model geometry.
- **Materials** ( ): Contains the materials and material properties used as sources for material data in the model. See Materials for detailed information.
- **Physics** ( ): Any added physics displays as a node under **Component** (**Solid Mechanics** in Figure 3-1 for example).
- **Multiphysics** ( ): When a multiphysics interface is added to the Model Builder, this node contains all the relevant multiphysics features for that interface. See Multiphysics Modeling Approaches for more information.
- **Meshes** (△): Contains the sequences of mesh operations that defines the computational meshes for the model. When there is only one mesh in the model, its **Mesh** node appears directly under the **Component** node.

Branches and Subbranches in the Tree Structure

The settings window has the following sections (also see Figure 3-3):

COMPONENT IDENTIFIER

The component identifier is a string used to define a namespace for the model component and identify variables defined in that component. The default component identifier is `comp1`, `comp2`, and so on, but you can change it in the **Identifier** field.

COMPONENT SETTINGS

This section contains general settings that you normally do not need to change:

Unit System

The default setting in the **Unit system** list, **Same as global system**, is to use the global unit system, which you specify in the root node's settings window. If you want to use another unit system in a model, select it from this list.

Spatial Coordinates

The default names for the spatial coordinates are x, y, and z for 3D as well as planar 1D and 2D geometries. For axisymmetric geometries, the default names for the spatial coordinates are r, φ (**phi**), and z. If you use the geometry to represent something other than space, or if you for some other reason want to use other names for the spatial coordinates, you can change the names in the fields for the **First**, **Second**, and **Third** coordinate under **Spatial coordinates**. The field labels include the default spatial coordinate names in parentheses.

You cannot use the variable for the time, t, as a spatial coordinate name.

Geometry Shape Order

The setting in the **Geometry shape order** list determines the order of the curved mesh elements that determine the geometry shape. The default setting is **Automatic**, but it is also possible to select an order such as **Linear**, **Quadratic**, **Cubic**, **Quartic**, **Quintic**, **Sextic**, and **Septic**. The default setting allows for automatic reduction of the order in some cases.

- Creating a New Model
- The Root Window
- Editing Node Properties, Names, and Identifiers
- Setting the Unit System for Models
- Using Extra Dimensions
- Curved Mesh Elements in the *COMSOL API Reference Manual*

Adding Extra Dimensions to a Component

To add an extra dimension to The Component Node, from the **Home** ribbon (Windows users), **Model Toolbar** (Mac and Linux users), or for any operating system, right-click the **Component** node and choose **Add Extra Dimension**. An **Extra Dimension** node, in the chosen space dimension, is then added under the Component node in the Model Builder. You can to add one or several extra dimension nodes to any Component.

The added node then contains these default nodes—**Definitions**, **Geometry**, and **Mesh**. The settings for the Extra Dimension are the same as for the Component node, except it has a unique identifier.

The default nodes associated to the Extra Dimension are considered the extra dimension geometry and extra dimension mesh. The original geometry and mesh are called the base geometry and base mesh.

Before the extra dimensions can be used in physics, they must be attached on a selection in the base geometry.

To enable the **Extra Dimensions** option, go to The Preferences Dialog Box. Click **Model Builder**, select the **Enable technology preview functionality** check box, and then click **OK**.

COMPONENT IDENTIFIER

The component identifier is a string used to identify variables in the model. The default Extra Dimensions component identifier is `xdim1`, `xdim2`, and so on, but you can change it in the **Identifier** field.

Using Extra Dimensions

Branches and Subbranches in the Tree Structure

You can proceed through your modeling in the Model Builder by selecting the branches in the order suggested by the default positions, from the top down, or selecting and defining each branch as needed. One level below the main Component branch are subbranches as described in Table 3-2 and shown in Figure 3-2. The node appearance can also change depending on many factors. See Dynamic Nodes in the Model Builder for examples.

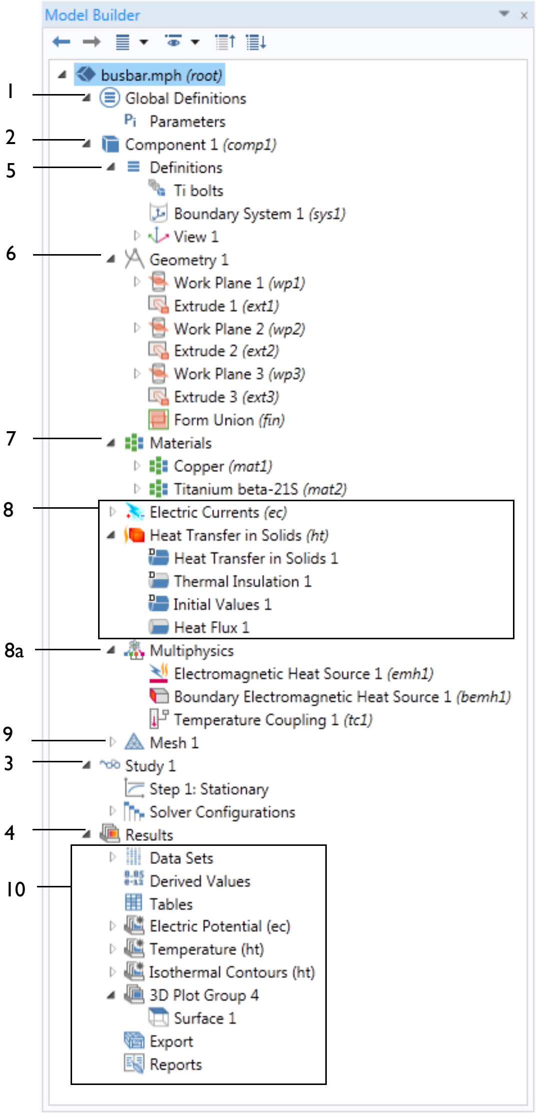


Figure 3-2: An example of the Model Builder tree structure showing the many different types of branches and subbranches available in a model. Refer to Table 3-2 to learn more about a node. Use the numbers to locate the node in the table.

TABLE 3-2: THE MODEL BUILDER BRANCHES AND SUBBRANCHES

FIGURE REF.	ICON	NAME	DESCRIPTION AND LINK TO MORE INFORMATION
Main Branches			
1	☰	Global Definitions	Has feature nodes to define global Parameters, Variables, Functions, Load and Constraint Groups, and Geometry Subsequences that are available everywhere in the model. See Global Definitions.
2	Various	Component	This branch includes the subbranches Definitions, Geometry, Materials, Physics, and Mesh. Also see The Component Node.

TABLE 3-2: THE MODEL BUILDER BRANCHES AND SUBBRANCHES

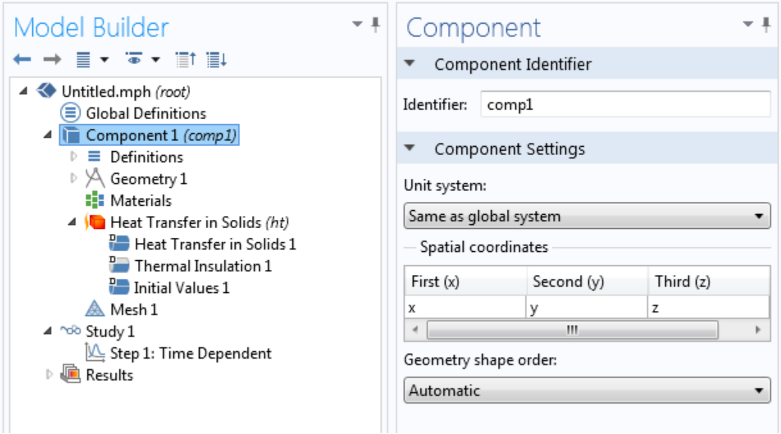
FIGURE REF.	ICON	NAME	DESCRIPTION AND LINK TO MORE INFORMATION
3		Study	This subbranch is where you set up study steps and solver configurations to solve a model using one or more study types for different analyses. See Studies and Solvers.
4		Results	The features contained in the subbranches for Data Sets, Derived Values, Tables, Export, and Reports are used to present and analyze results. See Results Analysis and Plots.
Subbranches			
5	≡	Definitions	This subbranch is used to create variables, functions, selections and other definitions that are local to a specific component in your model. Features available include Functions, Selections, Coordinate Systems, Component Couplings, and Probes.
6		Geometry	This branch contains the definition of the model's geometry, where you can import a geometry or build one yourself using the available tools. See Geometry Modeling and CAD Tools.
7		Materials	Collect all material properties organized in Material nodes with a defined geometric scope. Material properties required by any of the physics show up automatically in the defined material's settings window. See Materials.
8	Various	Physics	Each physics interface forms its own branch based on the model definition requirements. See The Physics Interfaces and Creating a New Model to start.
8a		Multiphysics	This is a main branch but is associated directly with the physics branch. See The Multiphysics Node.
9	△	Mesh	This subbranch collects all meshes defined for a model. If there is only a single mesh in a model, its Mesh node appears directly under the corresponding Component node. See Meshing.

TABLE 3-2: THE MODEL BUILDER BRANCHES AND SUBBRANCHES

FIGURE REF.	ICON	NAME	DESCRIPTION AND LINK TO MORE INFORMATION
10	▦	Data Sets	Data Sets refer to the source of data for creating Plots and Reports. It can be a Solution, a Mesh, or some transformation or cut plane applied to other data sets—that is, you can create new data sets from other data sets.
	8.85 e-12	Derived Values	Used to define evaluations of numerical results—globally, in a point, or integrated quantities. For 2D and 3D plots, you can also get numerical results directly in a table by clicking the plot. See About Derived Values.
	▦	Tables	This subbranch displays the results from integral and variable evaluations defined in Derived Values nodes or by probes and stored in Table nodes. See The Table Window and Tables Node.
	Various	Plot Groups	After adding a 3D, 2D, or 1D Plot Group, plots are added and defined under this subbranch. See Plot Groups and Plots.
		Export	After a model is completed, you can add various components to this and then generate outputs (animations, data, images, player, or export), or export the information to your computer as image, movie, or data files for use in external documents or for other purposes. See Export Types.
		Reports	This subbranch opens the Report Generator, which is a tool for reporting and documenting models created in COMSOL. It creates a record of the entire model including all the settings made during the modeling process. The report is an overview of the model and includes model properties, geometry, physics, mesh, studies, and results and visualization. See Reports.

The Node Settings Windows

For all operating systems, and when any node is clicked in the Model Builder (except a few *container nodes* such as Definitions and Data Sets), a corresponding settings window opens with the same name as the node. The settings window contains settings for defining operations and properties specific to a node, as shown in Figure 3-3.



Identifier: comp1

Unit system:

Same as global system

Spatial coordinates

First (x)	Second (y)	Third (z)
x	y	z

Geometry shape order:

Automatic

Figure 3-3: An example of a node settings window. In this example, the Component node settings window opens when the node of the same name is clicked.

When an operation or property is updated in the settings window, its effect on the model is displayed in the Graphics window either instantaneously or by clicking the **Build Selected** ( ), **Build All** ( ), or **Plot** ( ) button, which are available in some of the settings window toolbars. If you update settings for the physics, you must recompute the solution to reflect the changes in the physics.

To select the parts of the model to define in a specific settings window, select the relevant geometric entities directly in the displayed model in the Graphics window, from the Selection List window, or as, for example, **All domains** in the settings window.

- Toolbars and Keyboard Shortcuts
- About Selecting Geometric Entities
- The Graphics Window

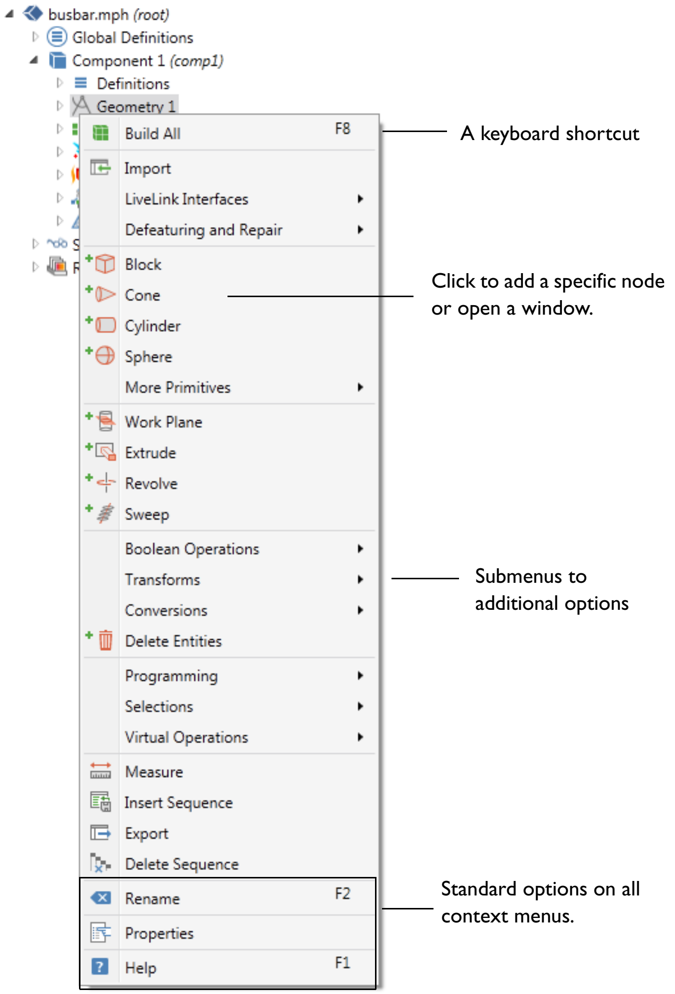
Opening Context Menus and Adding Nodes

In addition to using the toolbars and menus (see The COMSOL Desktop Menus and Toolbars), you can right-click a node to open a *context menu*. The context menu lists all the functionality available as subnodes to a particular node on a branch of the tree. Figure 3-4 shows the context menu for some of the Geometry node options. From the menu, you can add additional, and relevant, functionality, operations, or attributes to the sequence. Often there is a mixture of submenus, keyboard shortcuts, or specific features to choose from as in Figure 3-4 and Figure 3-5. There are also standard options such as **Rename**, **Properties**, and **Help**.

The context menu is also further divided and categorized for physics interfaces, as in the section Physics Node Context Menu Layout and Figure 3-5.

To add physics nodes to physics interfaces, in general, use the **Physics** toolbar. However, subnodes/subfeatures are added from the context menu.

The layout of the context menu (especially for physics interfaces) depends on whether the nodes are grouped by space dimension. The default is ungrouped nodes. See Grouping Nodes by Space Dimension and Type for an example comparing the different context menus.



A keyboard shortcut

Click to add a specific node or open a window.

Submenus to additional options

Standard options on all context menus.

Figure 3-4: A context menu opens when you right-click any node in the Model Builder. In this example, the options available for the Geometry node are shown.

OPENING THE CONTEXT MENU

- Right-click any node in the **Model Builder** to open the context menu
- Once a node is highlighted, right-click anywhere in the **Model Builder** to open it.
- Use the shortcuts based on operating system:
  - Windows: Press Shift+F10.
  - Mac: Press Ctrl+F10.
  - Linux: Press Shift+F10.

After selecting such an option from the list, an associated settings window opens to the right (by default) of the **Model Builder** window. See Figure 3-3 for an example.

In the context menu, a plus sign next to the icon (+▦) means a node of that type is added to the Model Builder.

- The Node Settings Windows
- Grouping Nodes by Space Dimension and Type
- Clearing Sequences and Disabling, Enabling, and Deleting Nodes
- About Geometric Entities

The Physics Nodes

An important part of building a model is where you add physics branches. For example, when Creating a New Model. This branch (see Figure 3-2 for an example) contains the nodes that define the material properties, equations, loads, initial values, boundary conditions, and other parts of the physics that the model describes. All settings windows for the specification of the physics and equations accept parameters and variables as input data.

SPECIFYING PHYSICS INTERFACE SETTINGS

Each physics interface includes nodes for specifying all input data for a specific physics in a model:

- Material properties and material models
- Boundary and interface conditions
- Equations (for equation-based modeling)
- Initial conditions

In addition, you can specify weak form contributions and element types for additional flexibility.

Specifically, the settings are available on the following parts of the geometry:

- Domains
- Boundaries
- Edges
- Points
- Additional properties that are independent of the geometry

Not all of these options are available for all geometry types and physics interfaces.

PHYSICS NODES BY SPACE DIMENSION

The physics nodes indicate the geometric entity level (domains, boundaries, edges, points, or pairs) based on the space dimension of the Component (see Table 3-3). The nodes also correspond to The Graphics Window Toolbar Buttons, some of which are also based on space dimension.

See Physics Exclusive and Contributing Node Types and Physics Node Status for examples of other differences to how the nodes display in the Model Builder.

TABLE 3-3: PHYSICS NODES BY SPACE DIMENSION

NAME	3D	2D AND 2D AXISYMMETRIC	1D AND 1D AXISYMMETRIC
Domain level	🟦	🟪	─
Domain level, default node	D 🟦	D 🟪	D ─
Boundary level	▤	⊖	─▪
Boundary level, default node	D ▤	D ⊖	D ─▪
Boundary level, Pairs	▤ ⋈	⊖ ⋈	─▪ ⋈
Point level	⬚	⬚	—
Edge level	⊟	—	—

- The Physics Interfaces
- Physics Default Nodes
- Physics Node Context Menu Layout
- Physics Node Status

Physics Default Nodes

When you add a physics interface, COMSOL automatically adds a corresponding physics branch in the tree, which typically includes a number of default nodes, including but not limited to:

- A model equation or material model node, typically on the domain level. This node defines the domain equations (except optional sources, loads, reactions, and similar contributing domain quantities) and the related material properties or coefficients.
- A boundary condition node. For multiphysics interfaces there is one boundary condition for each physics.
- For axisymmetric models, the symmetry axis has an **Axial Symmetry** boundary condition (see Physics Axial Symmetry Node).
- An **Initial Values** node for specifying initial values for a time-dependent simulation or an initial guess for the solution to a nonlinear model.

In most cases, the default nodes' initial selections include all domains or all boundaries (or all instances of another geometric entity level). Their selection is always every instance that is not overridden by another node on the same geometric entity level. It is not possible to delete such default nodes, but you can copy and duplicate all default nodes. Some multiphysics interfaces also add default nodes with no initial selection, which are possible to delete from the model. Default nodes include a *D* (for “default”) in the upper-left corner ( ) to indicate their special status. The copy or duplicate of a default node is a node of the same type but behaves as a normal node with an initially empty selection.

For example, for a geometry with four boundaries, the default boundary condition’s initial selection includes all four boundaries. If another exclusive boundary condition for Boundary 3 is added, that boundary becomes overridden (inactive) in the default boundary condition’s selection. If you disable or remove that boundary condition, the default boundary condition becomes active for Boundary 3 again. You cannot change a default node’s selection.

Some physics also add standard nodes directly when you add them to a model. They represent functionality that is likely to be useful but that you might want to make only active on a part of the geometry or delete. Such nodes do not include a *D* in the upper-left corner.

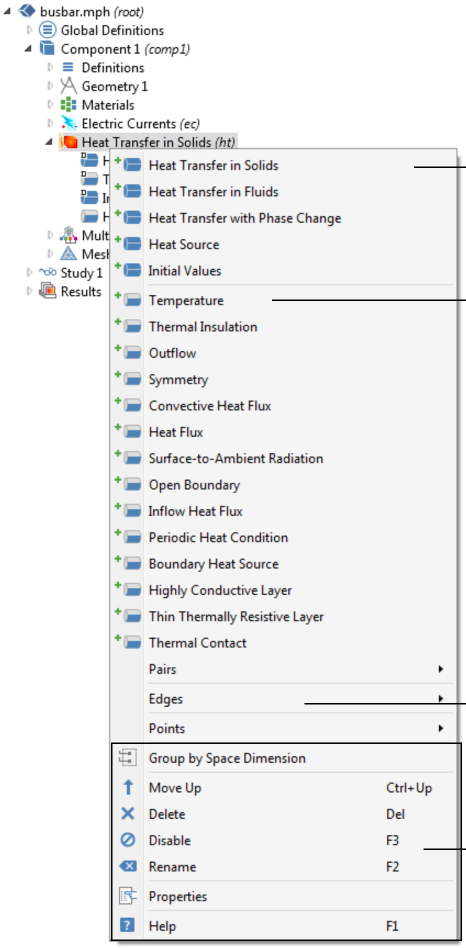
- Physics Nodes by Space Dimension
- Physics Node Status
- Physics Exclusive and Contributing Node Types
- Dynamic Nodes in the Model Builder

Physics Node Context Menu Layout

The context menu opens when you right-click a physics interface node, or any node in the Model Builder (see Opening Context Menus and Adding Nodes). This menu is divided into four sections for most physics: the first section contains domain settings, the second boundary settings, the third edge settings, and the fourth has point settings.

There can be menu items with the same name but applied at different geometric entity levels.

To add physics nodes to physics interfaces, in general, use the **Physics** toolbar. However, subnodes/subfeatures are added from the context menu.



An example of the context help menu when the nodes are not grouped by space dimension (the default).

Domain level settings for ungrouped nodes

Boundary level settings for ungrouped nodes

Edges and Points submenus

Common options on physics interface context menus.

Figure 3-5: An example of a Heat Transfer in Solids interface context menu. The choices are based on the Component dimension as well as the physics interface. The menu is further divided by geometric entity level (domain, boundary, edge, point, and pair).

As shown in Figure 3-5, the context menu layout is also based on whether the nodes are not grouped (the default) or if **Group by Space Dimension** is selected.

Physics Exclusive and Contributing Node Types

The nodes for the physics are in a sequence, which acts like a macro that the software runs in a top-down order. Depending on the selection for each node, a node can totally or partially override, or shadow, a node earlier in the sequence. How the software treats these nodes depends on their relationship. There are two different types of nodes: exclusive and contributing as shown in Figure 3-6.

The exclusive and contributing nodes maintain the described behavior only in relation to similar types of nodes within the same physics (for example, you can have a temperature constraint and a pressure constraint for the same boundary in the same model).

What the node looks like in the Model Builder is based on the space dimension. See Physics Nodes by Space Dimension.

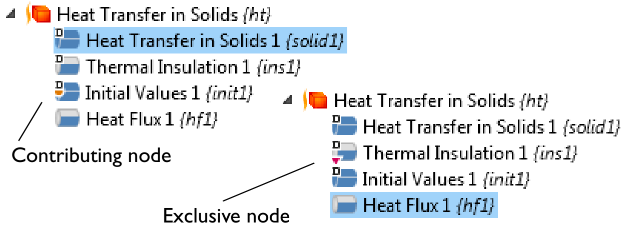

Heat Transfer in Solids {ht}
Heat Transfer in Solids 1 {solid1}
Thermal Insulation 1 {ins1}
Initial Values 1 {init1}
Heat Flux 1 {hf1}
Contributing node
Heat Transfer in Solids {ht}
Heat Transfer in Solids 1 {solid1}
Thermal Insulation 1 {ins1}
Initial Values 1 {init1}
Heat Flux 1 {hf1}
Exclusive node

Figure 3-6: An example of exclusive and contributing nodes in a 3D component's physics interface operating sequence and when Ungroup is selected.

EXCLUSIVE NODES

The use of an *exclusive node* means that only one can be active for a given selection. That is, if you add another exclusive node (for example, an identical node) with the same selection, the first exclusive node is overridden and thus has no effect.

Typical exclusive nodes include model equations, initial values, and boundary conditions that are constraints, such as prescribed values for displacements, temperatures, pressures, and so on, including special variants of these such as ground conditions in electromagnetics and fixed constraints in structural mechanics. Also some boundary conditions that are not constraints but have a definitive meaning are exclusive nodes—for example, electric insulation, thermal insulation, and no-flow conditions. Depending on the selections for each node, an exclusive node can override another node partially. Nodes are exclusive only within their specific physics. When a node is selected in the Model Builder tree, nodes that are overridden by the selected node are indicated using a red arrow in the lower-left corner of the icon ( ), and nodes that override the selected node are indicated using a red arrow in the upper-left corner of the icon ( ).

CONTRIBUTING NODES

A *contributing node* means you can have more than one of these nodes with the same selection and that the software adds these together when evaluating the model. Typical contributing nodes are loads, fluxes, and source terms, where you can have more than one of each type that is active on the same domain or boundary, for example. The total effect is then a sum of each contributing load, for example. When a node is selected in the Model Builder tree, COMSOL shows other nodes, which the current node contributes with, indicated using a yellow dot to the left of the icon (for example, in this boundary level icon). See also Figure 3-6 for an example.

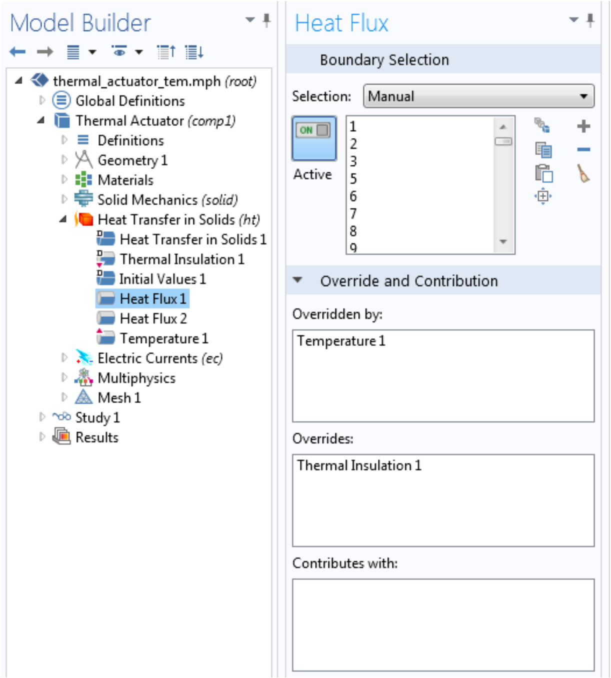
LISTING OVERRIDES AND CONTRIBUTIONS

If your preferences include showing the **Override and Contribution** section in the settings windows for physics nodes, you can find the following information about how exclusive and contribution nodes interact in the model. Click the **Show** button ( ) and select **Override and Contribution** from the **Model Builder** to display the section. To expand this section on all nodes, click **Expand Sections** ( ) and select **Override and Contribution** from the **Model Builder** and click the physics nodes to display the information as in Figure 3-7 and described below.

- The **Overridden by** list contains the names of the nodes that the selected node are overridden by. The selected node is then overridden by these nodes at least partially, and the **Selection** list contains **(overridden)** for the geometric entities (boundaries, for example) where it is overridden. The nodes that the selected node is overridden by are indicated using a red arrow in the lower-left corner of the icon, for example, in this boundary level icon .

- The **Overrides** list contains the names of the nodes that the selected node overrides (where the current node is active). The nodes that the selected node overrides are indicated using a red arrow in the upper-left corner of the icon, for example, in this boundary level icon .
- The **Contributes with** list contains the names of the nodes that the selected node contributes with for at least some shared selection. The nodes that the selected node contributes with are indicated using a yellow dot to the left of the icon, for example, in this boundary level icon .

If you disable physics nodes locally in a study step using the **Physics and Variables Selection** section in the study step's settings window, the indications of overrides and contributions in the Model Builder are unchanged (but disabled physics nodes get an asterisk to indicate that their state has been changed in at least one study step). The local variables and physics tree in the study step's settings window, on the other hand, does display the overrides and contributions taking the disabled nodes into account.



Selection: Manual

Overridden by:

Temperature 1

Overrides:

Thermal Insulation 1

Contributes with:

Figure 3-7: The Override and Contribution section lists other physics nodes that the selected node is overridden by, overrides, or contributes with. This is an example for a 3D component and with Ungroup selected.

- Physics Node Status
- Physics Default Nodes

Physics Node Status

The status of a physics node depends on if it is a default node, the selection that it applies to, and other nodes in the same branch that can override nodes earlier in the sequence. You can change the order of nodes (except the default nodes) by moving them up or down.

OVERRIDDEN SELECTIONS

A node can be partially or completely *overridden* by another node further down in the same branch of the model tree that is of a similar, exclusive type. For example, if you specify a temperature boundary condition on boundary 1 and boundary 3, and then add another temperature boundary condition for boundary 3, the first temperature boundary condition is overridden on boundary 3. In the settings window for the **Temperature** nodes that define the temperature boundary condition, the **Selection** list then shows **3 (overridden)** to indicate that the temperature boundary condition defined on this selection is overridden for boundary 3 but is still active on boundary 1. Deleting or disabling the other temperature boundary condition on boundary 3 reactivates the original temperature boundary condition, and then shows **3** (without the **(overridden)** indication).

SELECTIONS THAT ARE NOT APPLICABLE

For selections that are not applicable for a node (such as interior boundaries for an boundary condition that is only applicable for exterior boundaries), the **Selection** list then shows **(not applicable)** next to entries that are, in this case, interior boundaries.

ENABLING AND DISABLING NODES

By enabling or disabling physics nodes, you can activate and inactivate (shadow) other physics interface nodes that appear higher up in the physics inter face branches.

- Physics Default Nodes
- Physics Exclusive and Contributing Node Types
- Physics Nodes by Space Dimension
- Clearing Sequences and Disabling, Enabling, and Deleting Nodes

Dynamic Nodes in the Model Builder

The Model Builder is a dynamic environment. As your model is built and analyzed, there are numerous ways to quickly identify nodes that change status during the process. Table 3-4 lists generic examples and links to the dynamic visual aids that are used to help you.

- Branches and Subbranches in the Tree Structure
- The Component Node

TABLE 3-4: DYNAMIC NODES—VISUAL AIDS TO IDENTIFICATION

ICON	TYPE	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
✖	Error	For example, on a Material node. See Errors Relating to the Material Nodes.
⊗	Error node	Errors and Warnings

TABLE 3-4: DYNAMIC NODES—VISUAL AIDS TO IDENTIFICATION

ICON	TYPE	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
□	Current node, not built (yellow frame)	For example, on a Geometry node ( ). This node is also displaying the asterisk indicating the node is being Edited. The asterisk also appears on plot nodes when the plot has not been updated to reflect changes in the data or settings (for example, after re-solving), See The Current Node in Geometry Sequences.
□	Current node (green frame)	A current node is used for Geometry and Meshing nodes and indicates that the feature or sequence of steps has been built. It is a green line on the left and upper edges of the node. For example, on a Geometry node , after building. Also see The Current Node in Geometry Sequences.
▨	Enabled sequence	During solution processing, the particular sequence that is enabled and runs when selecting Compute has a green border around its icon ( ). See Computing a Solution.
~	Harmonic Perturbation	For example, on a boundary level node for the Electric Currents interface, Electric Ground node . See Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.
⚠	Warning	For example, on a Mesh node .
*	Editing, or in process of editing, a node	For example, on a Mesh node . This node is also displaying the asterisk indicating the node is being Edited. Also indicates physics nodes that have been disabled in a Study Step. See Editing and Building Geometry Nodes for Geometry nodes for example.
⊡	Pairs	For example, on a 3D Boundary Level node . See Identity and Contact Pairs.
▯ ▯	Pairs—Fallback Features	For example, on a 3D Boundary Level pair node . See Identity and Contact Pairs.
●	Contributing node	For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types and Physics Node Status.
D	Default node	For example, on a 2D boundary level node . See Physics Default Nodes.
▲	Override	For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types.
▼	Overridden	For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types
STUDY STEPS ANALYSIS		
⊙	Solve For	For example, a Laminar Flow interface where the green dot in the lower-right corner indicates that the study solves for the degrees of freedom in this physics interface. See Physics and Variables Selection.

TABLE 3-4: DYNAMIC NODES—VISUAL AIDS TO IDENTIFICATION

ICON	TYPE	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
◉	Disable in Solvers	For example, a Laminar Flow interface is enabled (not dimmed), shows that the study step provides degrees of freedom (the yellow dot in the lower-right corner), and has a change of state indicated by the asterisk. The yellow dot means that the study step provides degrees of freedom but does not solve for the physics. See Physics and Variables Selection.
*	Change of State (editing)	An asterisk appears in the upper-right corner of nodes for which you change their state in the study step's selection tree compared to their state in the main model tree in the Model Builder. For example, for the Joule Heating interface .
◉	Disabled in Model (provides no degrees of freedom) and shows a change of state	In this example, a Transport in Diluted Species interface is disabled (unavailable), provides no degrees of freedom (red dot in the lower-right corner), and has a change of state indicated by the asterisk. See Physics and Variables Selection.
LOAD AND CONSTRAINT GROUPS		
	Load Group	This is an example of a Boundary Load node with a load group . This is for a 2D model at the boundary level. See Load Group and Using Load Cases.
	Constraint Group	This is an example of a Fixed Constraint node with a constraint group . This is for a 2D model at the boundary level. See Constraint Group and Using Load Cases.

Errors and Warnings

COMSOL Multiphysics reports problems of two types—errors and warnings.

Errors: Errors prevent the program from completing a task. For errors, a **COMSOL Error** window appears with a brief error description and, in some cases, an **Open log file** button for additional information. Under the node where the error occurred there is, in most cases, also an **Error** subnode (⊗) that contains an error message that generally provides additional information. Also, for many error types, the icon for the node where the error occurred appears with a red cross in the lower-right corner.

Warnings: Warnings are problems that do not prevent the completion of a task but that might affect the accuracy or other aspects of the model. Warnings typically appear in the **Log** window (▦). The warning message also appears as a **Warning** subnode (⚠) under the node from which the warning was sent.

INDICATION OF UNEXPECTED, UNKNOWN, OR INCONSISTENT UNITS

The unit display appears orange for the properties in the settings for the physics and materials that have invalid or inconsistent units or a different unit than expected. An inconsistent unit can occur by summing terms with units that represent different physical quantities, such as `273[K]+3[ft]`, for example. A tooltip displays a message at the corresponding field. In the case of a valid but unexpected unit, this message contains the deduced and expected units in the current unit system.

If an unexpected or inconsistent unit appears in a text field for a physical property, COMSOL ignores the unit and uses the numerical value, including an SI prefix if present, as the input to the model. For example, in a text field for density using SI units, the software interprets `2930[K]` as 2930 kg/m^3 and `2930[mK]` as 2.930 kg/m^3. A unit display that appears red contains a syntax error, which can be due to, for example, missing or misplaced parentheses.

ERRORS AND WARNINGS IN A GEOMETRY SEQUENCE

If an error occurs when you build a node, the build stops. The node with the problem then gets an **Error** subnode (⊗) that contains the error message. Also, the node's icon displays with a red cross in the lower-left corner.

After a successful build of a node, a warning message can sometimes display as a **Warning** subnode (⚠). If a warning message exists, the node's icon displays with a yellow triangle in the lower-left corner.

ERRORS AND WARNINGS IN MESHING SEQUENCES

If a problem occurs when you build a node, the build continues if it is possible to avoid the problem in the corresponding meshing operation, otherwise the build stops.

Continuing the Build

When you build a Free Triangular, Free Quad, or Free Tetrahedral node where problems are encountered, you can avoid the problems related to meshing of faces and domains by leaving the corresponding faces and domains unmeshed. The operation corresponding to the node continues meshing the remaining entities and stores information on the encountered problems in subnodes of the node. A node that encountered this type of problems during the build gets a Warning status. The node's icon is decorated with a yellow triangle in the lower-right corner. If you build several nodes in a sequence the build does not stop by a node that gets a Warning status. See Dynamic Nodes in the Model Builder.

Stopping the Build

When you build other nodes than Free Triangular, Free Quad, or Free Tetrahedral the build stops if a problem is encountered. This means that no changes are made to the mesh. The node gets an Error status, which the program indicates by adding a red cross in the lower-right corner of the node's icon. You find information about the error in a subnode of the node where the error occurred. If the node is part of a sequence build, the build stops and the preceding node becomes the current node.

ERRORS AND WARNINGS IN SOLVER SEQUENCES

Problems encountered when running a solver or generating a mesh are treated in two different ways depending on if it is possible to avoid the problem and continue the operation or if the operation must be stopped. In the first case, a **Warning** node (⚠) appears under the node in the model tree that caused the problem. In the second case, an **Error** node (⊗) appears under the node in the model tree that caused the problem. For the solvers, and for multiple meshing warnings and errors, you find the **Warning** and **Error** nodes under the **Information** node (ⓘ).

- Using Units
- Unit Systems in COMSOL
- Dynamic Nodes in the Model Builder

Working with Nodes in the Model Builder

Viewing Node Names, Identifiers, Types, and Tags

Select the node labels to display in the Model Builder. You can view combinations of the following label types:

- *Names* are only used in the Model Builder for names (descriptions) of the nodes. You can rename the nodes, using any name that is unique within the model, to better describe what they do or contain in a model.
- *Tags* are unique for each node and are assigned automatically. Tags are primarily used when running COMSOL models in Java or MATLAB
- *Identifiers* are primarily used as names for functions and operators when called in models and as identifiers for the component and physics interface nodes that you use for variable scoping. See Variable Naming Convention and Namespace. You can change the name of the component and interface identifiers. An identifier must begin with a lowercase or uppercase letter (a–z or A–Z). All other characters in the identifier must be a lowercase or uppercase letter, a number 0–9, or an underscore (_).
- *Types* are the unique descriptions for each type of node, which you cannot modify.

You can change the name of a node for all levels (except the root node, which gets its name from the model filename), but the identifier can only be changed for the top **Component** and physics interface nodes and as a name for functions and operators. The tag and type are automatically assigned by the software and cannot be changed.

You can specify a node's name in its **Properties** section, by right-clicking the node and choosing **Rename**, or by pressing F2, which opens a window where you can enter a **New name**. The identifier serves as the name of functions and operators, which you can define in the settings windows' **Function Name** or **Operator Name** section for those nodes. You can also specify it in the **Component Identifier** and **Interface Identifier** sections for the main **Component** nodes and physics interface nodes, respectively.

Win

To access the menu where you select the node labels to display, on the **Home** ribbon, in the **Layout** section, click **Model Builder Node Label**.

Mac
Linux

To access the menu where you select the node labels to display, select a layout from **Windows>Model Builder Node Label**.

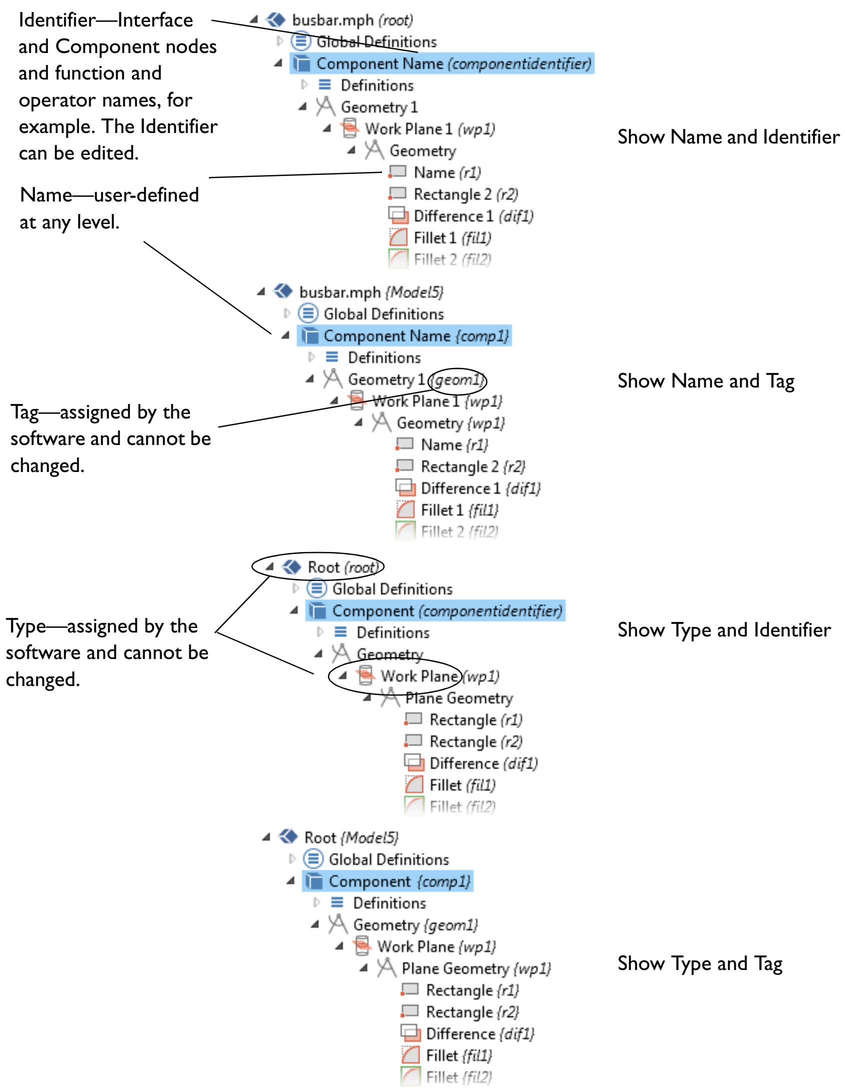
SELECTING THE LABELS TO VIEW IN THE MODEL BUILDER

Select from the list: **Show Name Only**, **Show Name and Identifier**, **Show Name and Tag**, **Show Type and Identifier**, or **Show Type and Tag**. See Figure 3-8 for examples of the different label types.

The options in the list modify the view in the **Model Builder** in the following way:

- Select **Show Name Only** to only display the name.
- Select **Show Name and Identifier** to display the name with the identifier in parentheses using an italic font. The identifier appears only where it is defined—that is, for component nodes, physics nodes, functions, component couplings, and geometry features. This is the default setting.

- Select **Show Name and Tag** to display each node's feature name with the predefined tag in curly braces using an italic font.
- Select **Show Type and Identifier** to display each node's feature type (predefined name) with the identifier in parentheses using an italic font.
- Select **Show Type and Tag** to display each node's feature type (predefined name) with the predefined tag in curly braces using an italic font.



Identifier—Interface and Component nodes and function and operator names, for example. The Identifier can be edited.

Name—user-defined at any level.

Tag—assigned by the software and cannot be changed.

Type—assigned by the software and cannot be changed.

Show Name and Identifier

Show Name and Tag

Show Type and Identifier

Show Type and Tag

Figure 3-8: Examples of the available label combinations on the View menu.

- Editing Node Properties, Names, and Identifiers
- The Root Window

Moving Nodes in the Model Builder

Many of the nodes under the branches and subbranches listed in Table 3-2 can be moved around in the model tree. To move nodes use one of these methods:

- Select the nodes and use the mouse to drop them in another applicable position in the model tree. A horizontal line indicates where in the model tree the moved (or copied) nodes get inserted when releasing the mouse.
- Right-click the selected nodes and select **Move Up** (↑) or **Move Down** (↓).
- Use the keyboard shortcuts Ctrl+up arrow or Ctrl+down arrow to move nodes up or down.

> For physics nodes it is not possible to move the default nodes (for the default boundary condition, for example). It is possible to create a copy of a default node, which initially has no selection. To click-and-drag a default node creates a copy whether or not the Ctrl key is pressed.

The order of the nodes in some of the branches affects the evaluation of the sequence that they define. In the following branches and subbranches it is possible to move nodes up and down to control the evaluation of the sequence or the order in which they appear within the branch or subbranch (also see Table 3-2):

- Definitions: nodes can be moved relative to other nodes of the same type (functions, selections, and so on).
- Geometry: Geometry nodes.
- Materials: Material nodes.
- Mesh: Mesh nodes.
- Physics interfaces: Except for the default nodes, the nodes for physics interfaces (such as material models, boundary conditions, domains, edges, points, and sources) can be moved within the physics branches.
- Study: The Study Step nodes can be moved.
- Results: The order of the nodes can be rearranged within each of the subbranches (Derived Values, Tables, Plot Groups, Export, and Reports). Exceptions under the Export node are the Plot, Mesh, and Table nodes.

Copying, Pasting, and Duplicating Nodes

It is possible to copy and paste many of the nodes in the Model Builder to create additional nodes with identical settings. Some nodes can also be duplicated underneath the original node. You can also move, copy, and duplicate nodes using “drag-and-drop” of nodes in the Model Builder.

> **Duplicate** (⧉) is a convenient way to copy and paste in one step. In other words, it combines the **Copy** and **Paste** functions. When a node is duplicated, COMSOL adds identical nodes underneath the original nodes on the same branch. Most, but not all nodes, can be duplicated.

Nodes that can be copied (and duplicated) include the following:

- Functions, which are possible to copy from one **Definitions** or **Global Definitions** branch to another. Also see Functions and About Global and Local Definitions.
- Physics nodes, which can be copied within the same physics interface or to another identical physics interface. Also see The Physics Interfaces.
- Geometry sequences—There are two ways to copy and paste geometry objects. Using the Transforms>Copy operation (that keeps the nodes linked to one another), or a standard copy and paste (see Copy and Paste

Geometry Objects). It is also possible to copy/paste and duplicate nodes corresponding to operation features, such as the Union node.

- Study steps, which are possible to copy from one **Study** branch to another. Also see Studies and Solvers.
- Plot nodes, which are possible to copy from one plot group to another. Also see Plot Groups and Plots.

> The copied object must be pasted into a model with the same space dimension. For example, a Sphere can only be pasted into a 3D model.

HOW TO COPY, PASTE, OR DUPLICATE NODES

- On the Quick Access Toolbar (Windows users) or from the main **Edit** menu (Mac and Linux users), click **Copy** ( ), **Paste** ( ), or **Duplicate** ( ).
- Right-click a node and select **Copy**, **Paste**, or **Duplicate**.
- To paste a node, and after selecting **Copy**, click the parent node and right-click to select **Paste Heat Flux** to paste a copied node (a Heat Flux node in this case) to the parent node's branch.
- Create a copy of a node by Ctrl-clicking it and dragging a copy to an applicable location. A small plus sign at the cursor indicates that you drag a copy of the selected node.
- Ctrl-click and drag a duplicate to an applicable location. A small plus sign at the cursor indicates that you drag duplicates of the selected nodes.

Undoing and Redoing Operations

> Undo is not possible for nodes that are built directly, such as geometry objects, solutions, meshes, and plots.

It is possible to undo the last operation for operations like adding, disabling, moving, and deleting nodes in the **Model Builder** as well as changing values in the settings window. You can undo or redo several successive operations.

To undo the last operation or redo an undone operation:

- On the Quick Access Toolbar (Windows users) or from the main **Edit** menu (Mac and Linux users), select or click **Undo** ( ) or **Redo** ( ).
- Press Ctrl+Z (undo) or Ctrl+Y (redo).

> - Copying, Pasting, and Duplicating Nodes
> - Clearing Sequences and Disabling, Enabling, and Deleting Nodes

Going to the Source Node

In the settings window for many nodes, other nodes can be referenced in the model tree such as a component, solution, study or study step, or data set, which provide data to the node where they are referenced.

Nodes where you refer to other nodes include plot groups, data sets, and solvers; in such nodes' settings windows, click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

- The Node Settings Windows
- Studies and Solvers
- Results Analysis and Plots

Clearing Sequences and Disabling, Enabling, and Deleting Nodes

You can change the contents, and actions, of the sequences in the model tree by clearing a mesh sequence or solution under a solver configuration. Or disable, enable, and delete nodes in the Model Builder.

> !
>
> **Undo** is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

CLEAR OR DELETE A MESH OR SOLUTION

Use a **Clear** function to keep the nodes and be able to recreate the mesh or solution by rebuilding the mesh sequence or computing the solution again.

- Under the **Component** node where you want to clear or delete the mesh, right-click the **Mesh** node and select **Clear Mesh** ( ). To delete a meshing sequence completely, select **Delete Sequence** ( ).
- To clear a set of solutions under a specific study, right-click the **Solver Configurations** node and select **Clear Solutions** ( ).To delete all **Solver** nodes, select **Delete Solvers** ( ). You can also choose whether or not to remove the Results nodes (data sets and plots, for example) associated with the solver configuration.

If you have a model geometry with several meshes or several studies, you can clear all meshes or all solutions in all studies at the same time. From the Mesh toolbar, click **Clear All Meshes** ( ). Or from the Study toolbar, click **Clear All Solutions** ( ).

DISABLE OR ENABLE NODES

A disabled node does not take part in the evaluation of a sequence; see Figure 3-4. Some nodes, such as container nodes and default nodes in the physics interfaces (see Physics Default Nodes), cannot be disabled (or deleted). When this is the case, the context menu does not have these options available. You can use Shift-click and Ctrl-click to select multiple nodes that you want to delete, disable, or enable.

- To disable selected nodes, right-click and select **Disable** ( ) or press F3. The nodes are unavailable (dimmed) in the model tree to indicate that they are disabled. For a geometry or mesh sequence, the disabled node does not affect the finalized geometry or mesh.
- To enable disabled nodes, right-click and select **Enable** ( ) or press F4.

> Instead of disabling and enabling variables and physics nodes to simulate different analysis cases (using different boundary conditions or sources, for example), use the selection of variables and physics in the study steps' **Physics and Variables Selection** sections, or use *load cases* for solving cases with varying loads or constraints. See Physics and Variables Selection and Using Load Cases.

DELETE NODES

- To delete selected nodes, right-click the nodes and select **Delete** (✖) or press Del (the Delete key). Confirm the deletion of nodes for it to take effect. Also see Clear or Delete a Mesh or Solution.
- To delete a geometry right-click the **Geometry** node in the **Model Builder** and select **Delete Sequence** ( ). You cannot use the **Undo** command.
- To delete geometry objects or entities, in the **Model Builder**, right-click **Geometry** and select **Delete Entities** (🗑). Or select objects in the **Graphics** window, and click the **Delete** button (✖) in the Graphics toolbar.

  If you use the **Delete** button to delete objects, COMSOL Multiphysics deletes the selected objects that correspond to primitive features by deleting their nodes from the geometry sequence. If you delete objects that do not correspond to primitive features or if you delete geometric entities a **Delete Entities** node appears in the sequence.

!

Undo is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

Modeling Guidelines

To model large-scale problems and for successful modeling in general, COMSOL makes it possible to tune solver settings and to use symmetries and other model simplifications to reach a solution or—failing that—interrupt the solution process to retrieve a partial solution. This section provides some tips and guidelines when modeling.

Selecting Physics

When creating a model in COMSOL Multiphysics, you can select a single physics interface that describes one type of physics or select several physics interfaces for multiphysics modeling and coupled-field analyses.

MODELING USING A SINGLE PHYSICS

Most physics interfaces contain Stationary, Eigenvalue, and Time Dependent (dynamic) study types. As already mentioned, these physics provide features and windows where you can create models using material properties, boundary conditions, sources, initial conditions, and so on. Each physics interfaces comes with a template that automatically supplies the appropriate underlying PDEs.

If you cannot find a physics interface that matches a given problem, try one of the interfaces for PDEs, which makes it possible to define a custom model in general mathematical terms. Indeed, COMSOL can model virtually any scientific phenomena or engineering problems that originate from the laws of science.

MULTIPHYSICS MODELING USING MULTIPLE PHYSICS INTERFACES

When modeling real-world systems, you often need to include the interaction between different kinds of physics: ***multiphysics***. For instance, an electric current produces heat, and the properties of an electronic component such as an inductor vary with temperature. To solve such a problem, combine two or several physics into a single model using the multiphysics capabilities of COMSOL. For the example just mentioned, you can use the predefined Joule Heating interface, which is a combination of the Electric Currents and Heat Transfer interfaces. This way you create a system of two PDEs with two dependent variables: V for the electric potential and T for the temperature. There are many other predefined multiphysics couplings that provide a unified physics interface that combines two or more coupled physics for common multiphysics applications.

You can also combine physics interfaces and equation-based modeling for maximum flexibility.

To summarize the proposed strategy for modeling processes that involve several types of physics: Look for physics interfaces suitable for the phenomena of interest. If you find them among the available physics interfaces, use them; if not, add one or more interface for equation-based modeling.

When coupling multiple physics in a multiphysics model (without using a predefined multiphysics interface), the couplings can occur in domains and on boundaries. COMSOL automatically identifies potential ***model inputs*** for quickly forming couplings between physics. For example, a velocity field from fluid flow is a model input for the convective heat transport in heat transfer. In that case, the model input automatically transfers the velocity field from the fluid to the heat transfer part.

Using Symmetries

By using symmetries in a model you can reduce its size by one-half or more, making this an efficient tool for solving large problems. This applies to the cases where the geometries and modeling assumptions include symmetries.

The most important types of symmetries are axial symmetry and symmetry and antisymmetry planes or lines:

- *Axial symmetry* is common for cylindrical and similar 3D geometries. If the geometry is axisymmetric, there are variations in the radial (r) and vertical (z) direction only and not in the angular (θ) direction. You can then solve a 2D problem in the *rz*-plane instead of the full 3D model, which can save considerable memory and computation time. Many physics interfaces are available in axisymmetric versions and take the axial symmetry into account.

- *Symmetry and antisymmetry planes or lines* are common in both 2D and 3D models. *Symmetry* means that a model is identical on either side of a dividing line or plane. For a scalar field, the normal flux is zero across the symmetry line. In structural mechanics, the symmetry conditions are different. *Antisymmetry* means that the loading of a model is oppositely balanced on either side of a dividing line or plane. For a scalar field, the dependent variable is 0 along the antisymmetry plane or line. Structural mechanics applications have other antisymmetry conditions. Many physics interfaces have symmetry conditions directly available as nodes that you can add to the model tree.

To take advantage of symmetry planes and symmetry lines, all of the geometry, material properties, and boundary conditions must be symmetric, and any loads or sources must be symmetric or antisymmetric. You can then build a model of the symmetric portion, which can be half, a quarter, or an eighth of the full geometry, and apply the appropriate symmetry (or antisymmetry) boundary conditions.

Effective Memory Management

Especially in 3D modeling, extensive memory usage requires some extra precautions. First, check that you have selected an iterative linear system solver. Normally you do not need to worry about which solver to use because the physics interface makes an appropriate default choice. In some situations, it might be necessary to make changes to the solver settings and the model. For details about solvers, see the Studies and Solvers chapter.

ESTIMATING THE MEMORY USE FOR A MODEL

Out-of-memory messages can occur when COMSOL tries to allocate an array that does not fit sequentially in memory. It is common that the amount of available memory seems large enough for an array, but there might not be a contiguous block of that size due to memory fragmentation.

In estimating how much memory it takes to solve a specific model, the following factors are the most important:

- The number of node points
- The number of dependent and independent variables
- The element order
- The sparsity pattern of the system matrices. The sparsity pattern, in turn, depends on the shape of the geometry and the mesh but also on the couplings between variables in a model. For example, an extended ellipsoid gives sparser matrices than a sphere.

The MUMPS and PARDISO out-of-core solvers can make use of available disk space to solve large models that do not fit in the available memory.

You can monitor the memory use in the bottom-right corner of the COMSOL Desktop, where the program displays the amount of physical memory and total virtual memory used.

CREATING A MEMORY-EFFICIENT GEOMETRY

A first step when dealing with large models is to try to reduce the model geometry as much as possible. Often you can find symmetry planes and reduce the model to half, a quarter, or even an eighth of the original size. Memory usage does not scale linearly but rather polynomially (Cn^k, $k > 1$), which means that the model needs less than half the memory if you find a symmetry plane and cut the geometry size by half. Other ways to create a more memory-efficient geometry include:

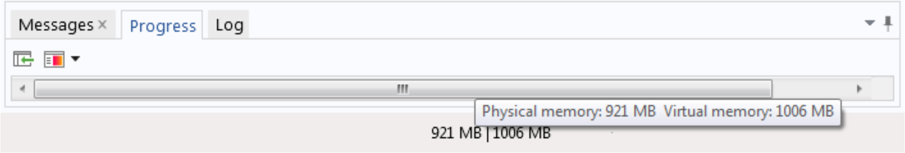
- Avoiding small geometry objects where not needed and using Bézier curves instead of polygon chains.

- Using linear elements if possible (this is the default setting in many physics interfaces). See Selecting an Element Type.
- Making sure that the mesh elements are of a high quality. Mesh quality is important for iterative linear system solvers. Convergence is faster and more robust if the element quality is high.
- Avoiding geometries with sharp, narrow corners. Mesh elements get thin when they approach sharp corners, leading to poor element quality in the adjacent regions. Sharp corners are also unphysical and can lead to very large (even infinite, in theory) stress concentrations.

INFORMATION ABOUT MEMORY USE

In the bottom-right corner of the COMSOL Desktop is information about how much memory COMSOL is currently using. The two numbers in Figure 3-9 displayed as **921 MB | 1006 MB** represent the physical memory and the virtual memory, respectively. If you position the cursor above these numbers, the tooltip that appears includes the numbers with the type of memory explicitly stated:

- The **Physical memory** number is the subset of the virtual address space used by COMSOL that is physically resident; that is, it is the amount of physical memory (RAM) in "active" use.
- The **Virtual memory** number is the is the current size of the virtual address space that COMSOL uses.



Physical memory: 921 MB Virtual memory: 1006 MB

921 MB | 1006 MB

Figure 3-9: An example of memory use displayed in COMSOL.

Selecting an Element Type

As the default element type for most physics, COMSOL uses first-order or second-order Lagrange elements (shape functions). Second-order elements and other higher-order elements add additional degrees of freedom on midpoint and interior nodes in the mesh elements. These added degrees of freedom typically provide a more accurate solution but also require more memory due to the reduced sparsity of the discretized system. For many application areas, such as stress analysis in solid mechanics, the increased accuracy of a second-order element is important because quantities such as stresses involve space derivatives and become constant within an element when using first-order elements.

COMSOL recommends that you use the default element types. For some applications, it might be possible to use a lower-order element than the default element type, but you must then use care to ensure that the important quantities are resolved.

For information about editing shape functions, see Equation View.

Analyzing Model Convergence and Accuracy

It is important that the numerical model accurately captures local variations in the solution such as stress concentrations. In some cases you can compare your results to values from handbooks, measurements, or other sources of data. Many model library examples are *benchmark models* that include comparisons to established results or analytical solutions.

If a model has not been verified by other means, a *convergence test* is useful for determining if the mesh density is sufficient. Here you refine the mesh and run the study again, and then check if the solution is converging to a stable value as the mesh is refined. If the solution changes when you refine the mesh, the solution is mesh dependent, so the model requires a finer mesh. You can use adaptive mesh refinement, which adds mesh elements based on an error criterion, to resolve those areas where the error is large. See the “Stresses and Strains in a Wrench” model in the *Introduction to COMSOL Multiphysics* book for an example for a convergence test.

For convergence, it is important to avoid singularities in the geometry.

Avoiding Singularities and Degeneracies in the Geometry

Achieving Convergence When Solving Nonlinear Equations

Nonlinear problems are often difficult to solve. In many cases, no unique solution exists. COMSOL uses a Newton-type iterative method to solve nonlinear systems of PDEs. This solution method can be sensitive to the initial estimate of the solution. If the initial conditions are too far from the desired solution, convergence might be impossible, even though it might be simple from a different starting value.

You can do several things to improve the chances for finding the relevant solutions to difficult nonlinear problems:

- Provide the best possible initial values.
- Solve sequentially and iterate between single-physics equations; finish by solving the fully coupled multiphysics problem when you have obtained better starting guesses.
- Ensure that the boundary conditions are consistent with the initial solution and that neighboring boundaries have compatible conditions that do not create singularities.
- Refine the mesh in regions of steep gradients.
- For convection-type problems, introduce artificial diffusion to improve the numerical properties. Most physics interfaces for modeling of fluid flow and chemical species transport provide artificial diffusion as part of the default settings.
- Scaling can be an issue when one solution component is zero. In those cases, automatic scaling might not work.
- Turn a stationary nonlinear PDE into a time-dependent problem. Making the problem time-dependent generally results in smoother convergence. By making sure to solve the time-dependent problem for a time span long enough for the solution to reach a steady state, you solve the original stationary problem.
- Use the parametric solver and vary a material property or a PDE coefficient starting from a value that makes the equations less nonlinear to the value at which you want to compute the solution. This way you solve a series of increasingly difficult nonlinear problems. The solution of a slightly nonlinear problem that is easy to solve serves as the initial value for a more difficult nonlinear problem.

- Stabilization Techniques
- Convergence Plots
- Introduction to Solvers and Studies

Avoiding Strong Transients

If you start solving a time-dependent problem with initial conditions that are inconsistent, or if you use boundary conditions or sources that switch instantaneously at a certain time, you induce strong transient signals in a system.

The time-stepping algorithm then takes very small steps to resolve the transient, and the solution time might be very long, or the solution process might even stop. Stationary problems can run into mesh-resolution issues such as overshooting and undershooting of the solution due to infinite flux problems.

Unless you want to know the details of the transients, start with initial conditions that lead to a consistent solution to a stationary problem. Only then turn on the boundary values, sources, or driving fluxes over a time interval that is realistic for your model.

In most cases, turn on your sources using a smoothed step over a finite time. What you might think of as a step function is, in real-life physics, often a little bit smoothed because of inertia. The step or switch does not happen instantaneously. Electrical switches take milliseconds, and solid-state switches take microseconds.

- Introduction to Solvers and Studies
- Stationary and Time Dependent

Physics-Related Checks and Guidelines

There are some important checks and guidelines that primarily apply to different areas of physics. Making these checks ensures that the model input is sufficient and increases the chances for successful modeling. See also the modeling sections of the documentation for the physics and the modules for more information related to modeling different physics.

FLUID FLOW AND TRANSPORT PHENOMENA

The following checks and guidelines primarily apply to fluid-flow modeling but also to modeling of other transport phenomena:

- If none of the boundary conditions include the pressure (most outlet conditions do, however), then you should specify the pressure at some point in the fluid domain. Without a specified pressure, the problem is underconstrained and it is difficult to get convergence.
- Make sure that the mesh if sufficiently fine, so that it contains at least 4–6 mesh elements across the thickness of a channel, for example.
- Make sure that the boundary conditions and the initial conditions match for time-dependent problems. For example, instead of starting with a full velocity on the wall, compared to a zero initial velocity field in the fluid, ramp up the velocity with a smoothed step function or a ramp function that takes the inlet velocity from zero, which matches the initial value for the velocity field, to the full velocity.

Avoiding Strong Transients

- For fluid-flow models it is important to estimate the flow regime (laminar or turbulent) using the Reynolds number, for example. If the flow is in the turbulent regime, a turbulence model is typically required.

ACOUSTIC, STRUCTURAL, AND ELECTROMAGNETIC WAVE PROPAGATION

For models that describe wave propagation, it is important to fully resolve the wave in both time and space. In practice that means using a maximum mesh element size that provides about 10 linear or five second-order elements per wavelength and also, for transient simulations, a fixed time step that is small enough.

STRUCTURAL MECHANICS

The following checks and guidelines primarily apply to modeling of structural mechanics:

- Make sure that the model is fully constrained. At a minimum, you typically need to constrain the model to avoid all rigid-body movement, which for a 3D solid mechanics model means 6 constraints for three translations and three rotations. Otherwise the solution is not well defined and does not converge. It is not possible to add all 6 constraints in a single point, where you can constrain at most three translational degrees of freedom. For a 3D solid model you can use a "3–2–1 approach" to constrain 3 degrees of freedom at one point (a fixed constraint), 2 at another point, and 1 at a third point. To do so, select three convenient points (vertices) that are well separated. Then fix the first point in all three directions. Constrain the second point in the two directions orthogonal (normal) to the vector from point one to point two making sure that there is no restriction to deformation along the line from point one to point two. Finally, constrain the third point in a direction normal to the plane formed by the three points. To test this approach, the model should expand or contract under temperature loading and have small stresses throughout with no stress concentrations. The corresponding minimum constraints for a 2D model are a fixed constraint at one point for the 2 translational degrees of freedom and an additional constraint in one direction at another point to constrain the single rotational degree of freedom.
- Consider if you can assume that the material is linear elastic and that the deformations are small. If not, consider using a nonlinear material model.
- Avoid sharp corners in the geometry, which are unphysical and lead to unbounded stress concentrations.

Results With Unphysical Values

WHERE AND WHY DO UNPHYSICAL VALUES APPEAR?

In some models small unphysical values can occur due to numerical artifacts or other model-related reasons. Examples include:

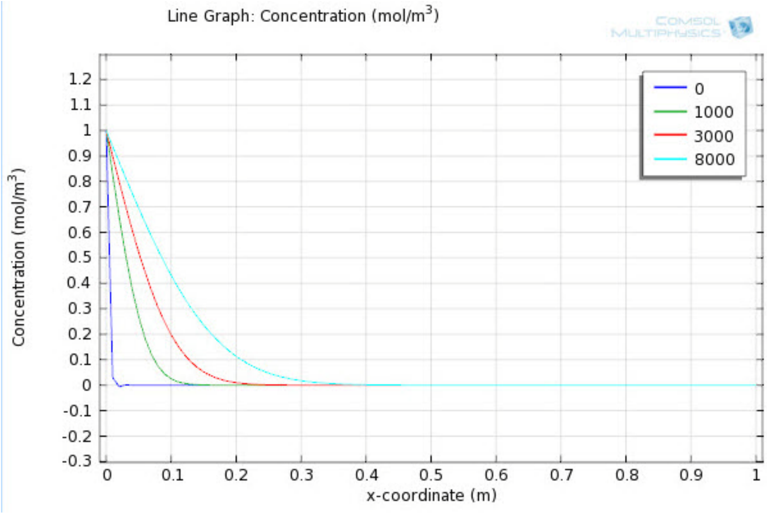
- Negative concentrations in mass transfer.
- A temperature that is slightly higher than the initial condition in time-dependent heat transfer studies.
- Small reaction forces that appear in unloaded directions in structural mechanics models.
- Small negative gaps in a contact analysis.
- Small negative effective plastic strain values.
- Stresses above the yield limit for an ideally plastic material in solid mechanics.

Some reasons for why these unphysical values occur:

- Numerical noise is a common cause. When the values of the dependent variables approach zero, the numerical noise can become relatively significant and cause some of the results to be slightly negative even if that is not physically possible.
- Interpolation and extrapolation of values can cause some values to become unphysical. Take care when using interpolated data or a piecewise polynomial function to define a temperature-dependent material property, for example. If you allow extrapolation outside of the defined range of input values, the material property values cannot be valid. Also, results for an elastoplastic material are correct (within some tolerance) at the integration points (Gauss points) inside the finite elements, but values might become unphysical when extrapolating the data to the element boundaries.

The Plasticity feature is available as a subnode to a Linear Elastic Material with the Nonlinear Structural Materials Module.

- Discontinuities in the model is another source of, for example, small negative concentrations due to a discontinuous initial value. With an initial value that is zero along a boundary for a convective transport models, for example, the physical interpretation is an initially sharp, gradually diffusing front moving away from the boundary. However, for the default shape function (second-order Lagrange elements), only continuous functions are admissible as solutions. COMSOL then modifies the discontinuous initial value before the time stepping can begin. This often results in a small dip in the solution at the start time. In the example model that the following figure shows, the concentration is locally slightly negative at $t = 0$:



- Lack of mesh resolution is another cause of unphysical values such as negative concentrations. The resulting convergence problems are often the underlying issue when negative concentrations are observed in high convection regimes (high Peclet number) and in those with large reaction terms or fast kinetics (high Damkohler number).
- Incorrect physics in the model can also cause these types of problems. For mass transfer, for example, the use of a constant sink in a reaction term is an approximation that only works for large concentrations. When the concentration reaches zero, the reaction term continues to consume the species, finally resulting in a negative concentration.

AVOIDING UNPHYSICAL VALUES

This section contains some ways to avoid computing or displaying unphysical values:

- In some cases it is possible to add a baseline to the dependent variable so that the numerical noise does not affect the solution in the same way as when the values of the dependent variable approach zero. This scaling is not possible with, for example, a reaction term that depends on the concentration because then the scale and origin do matter.
- Avoid discontinuities in the model using, for example, smoothed step functions.
- Formulate logarithmic variables as a way of eliminating mesh resolution problems and negative dips using the logarithm of the original dependent variable (the concentration, for example) as the dependent variable. The reason for this is that a linearly varying mesh sometimes does not capture the exponential behavior of the changes in the dependent variable. Modeling the logarithm of the dependent variable also ensures that the real concentration, for example, cannot become negative during the solution process.
- Avoid displaying small unphysical values due to numerical noise by clipping the values for the plot. You can do this by plotting, for example, `c*(c>0)` instead of `c`, which evaluates to 0 everywhere where c is smaller than 0. You can also adjust the range of the plot data and colors to only show nonnegative values. Parts of the plots where

values are outside the range then become empty.

- It can also be useful to check how the mesh affects the solution by refining the mesh and checking if the problem with unphysical values gets better or worse. If it gets better, then continue to refine the mesh. If it gets worse, you probably need to check the physics of the model.

Multiphysics Modeling Approaches

The ability to create multiphysics models—those with more than one type of physics or equation such as coupled-field problems—is one of the most powerful capabilities of COMSOL Multiphysics. In such a model, the software can solve all the equations, taken from various areas of physics, as one fully coupled system.

Within COMSOL Multiphysics you can choose from several ways to approach multiphysics modeling and coupled-field analysis.

In this section:

- Creating a Multiphysics Coupling
- Advantages of Using the Multiphysics Coupling Features
- The Multiphysics Node
- Uncoupling a Multiphysics Coupling
- Model Inputs and Multiphysics Couplings

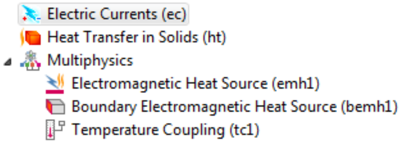
Creating a Multiphysics Coupling

There are two fundamental ways to create and use multiphysics couplings—using Predefined Multiphysics Couplings or by Adding Physics Sequentially.

PREDEFINED MULTIPHYSICS COUPLINGS

The Joule Heating Interface is a predefined multiphysics coupling. After **Joule Heating** is chosen from The Model Wizard, the **Heat Transfer in Solids** interface, the **Electric Currents** interface, and a **Multiphysics** node, including the default features applicable to the coupling (**Electromagnetic Heat Source**, **Boundary Electromagnetic Heat Source**, and **Temperature Coupling**), are displayed under the **Added physics** list as in Figure 3-10. Figure 3-11 shows you what is included in the Model Builder when a predefined multiphysics interface is added. Compare to Figure 3-12 where individual physics interfaces are added, and these features are initially accessible only from the context menu.

> You can add physics interfaces when you start creating the model with The Model Wizard or at any time with The Add Physics Window.



Electric Currents (ec)
Heat Transfer in Solids (ht)
Multiphysics
Electromagnetic Heat Source (emh1)
Boundary Electromagnetic Heat Source (bemh1)
Temperature Coupling (tc1)

Figure 3-10: When Joule Heating is selected in the Model Wizard, the default physics interfaces and coupling features are displayed under Added physics.

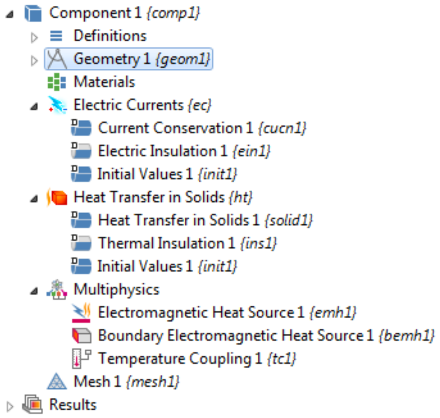

Component 1 {comp1}
Definitions
Geometry 1 {geom1}
Materials
Electric Currents {ec}
Current Conservation 1 {cucn1}
Electric Insulation 1 {ein1}
Initial Values 1 {init1}
Heat Transfer in Solids {ht}
Heat Transfer in Solids 1 {solid1}
Thermal Insulation 1 {ins1}
Initial Values 1 {init1}
Multiphysics
Electromagnetic Heat Source 1 {emh1}
Boundary Electromagnetic Heat Source 1 {bemh1}
Temperature Coupling 1 {tc1}
Mesh 1 {mesh1}
Results

Figure 3-11: An example of what is added to the Model Builder when Joule Heating is chosen in the Model Wizard. The Electromagnetic Heat Source, Boundary Electromagnetic Heat Source, and Temperature Coupling features are automatically included under the Multiphysics node.

ADDING PHYSICS SEQUENTIALLY

An empty **Multiphysics** node is added automatically when two (or more) physics interfaces are set up in a model and when there is the possibility to couple the interfaces. In other words, if you add physics interfaces one at a time, and the software identifies these interfaces as being of the multiphysics category, the Multiphysics node is automatically added to the Model Builder tree. The relevant features are made available from the context menu (right-click the Multiphysics node). See Figure 3-12.

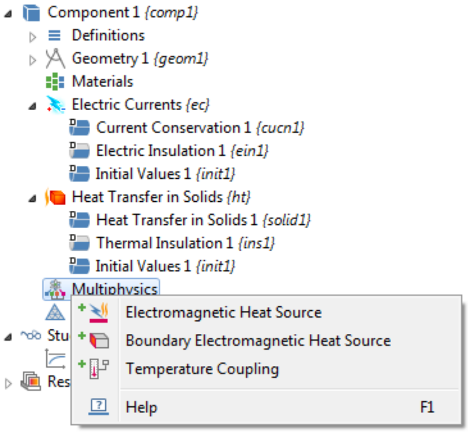

Component 1 {comp1}
Definitions
Geometry 1 {geom1}
Materials
Electric Currents {ec}
Current Conservation 1 {cucn1}
Electric Insulation 1 {ein1}
Initial Values 1 {init1}
Heat Transfer in Solids {ht}
Heat Transfer in Solids 1 {solid1}
Thermal Insulation 1 {ins1}
Initial Values 1 {init1}
Multiphysics
Stu
Res
Electromagnetic Heat Source
Boundary Electromagnetic Heat Source
Temperature Coupling
Help
F1

Figure 3-12: An example of when a Multiphysics node is automatically added to the model. The specific multiphysics features are made available from the context menu based on the physics interfaces in the model. The difference when the predefined Joule Heating interface is added is that these features are included under Multiphysics and there are some modified settings automatically applied. In either case, the available features depend on the add-on module license.

Advantages of Using the Multiphysics Coupling Features

One advantage to using the predefined couplings is that specific or modified settings are included with the physics interfaces and the coupling features. But if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single Electric Currents and Heat Transfer in Solids interfaces are added to the Model Builder, COMSOL adds an empty Multiphysics node. The features are available and can be added, but any modified settings are not included.

In general, it is useful to use any type of multiphysics coupling because you can turn multiphysics on and off (that is, enable and disable features), giving you more flexibility to test and observe multiphysics effects.

Even if you do not start with a predefined coupling, another benefit of this approach is that you are no longer constrained by the use of specific physics interfaces, nor do these have to be added in any specific order. With the new coupling approach, the order in which physics interfaces are added does not matter for the end result.

An example of this is if you start modeling by adding a Heat Transfer in Solids interface. As you continue to build the model, you add an Electric Currents interface. At this stage of the process, you have defined several boundary conditions, chosen materials, or experimented with other settings. You may have also solved the model successfully at this point and now you want to continue building on this design.

Previously, you would have to start again to add and define the Joule Heating interface to access the multiphysics features (Electromagnetic Heat Source, Boundary Electromagnetic Heat Source, and Temperature Coupling). COMSOL Multiphysics now recognizes this and automatically adds a Multiphysics node where you can right-click the Multiphysics node to access and add these features.

For multiphysics interfaces that consist of participating physics interfaces, the default solver settings use a segregated solver approach with one segregated step for each physics and each of these steps calling an iterative solver. These solver settings are suitable for large models, but if possible, a fully-coupled solver approach using direct solvers can be more robust. You can switch to such solver settings by right-clicking the **Study** node and choosing **Show Default Solver**. Then the solver nodes that the multiphysics interface specifies appear under the **Solver Configuration** node, and you can right-click the solver node to add a **Fully Coupled** solver node to replace the **Segregated** node, for example.

! For some multiphysics interfaces, a side effect of adding physics interfaces one at a time is that two study types—Frequency-Stationary and Frequency-Transient—are not available for selection until *after* at least one coupling feature is added. In this case, it is better to first add an **Empty Study**, then add the coupling features to the **Multiphysics** node, and lastly, right-click the **Study** node to add the study steps as required.

The Multiphysics Node

The **Multiphysics** node ( ) contains, or has available, any coupled physics features that are likely to be used for a particular set of physics interfaces added to the Model Builder. There are no settings required for the node itself.

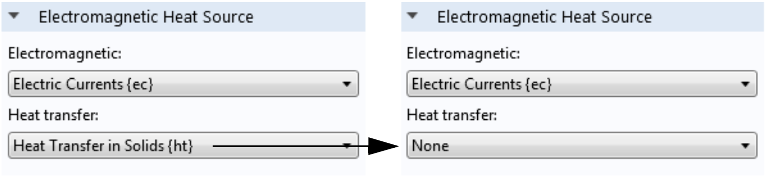
There are different approaches to the use of the multiphysics coupling feature. It is either predefined when you add a specific physics interface or it is automatically added when COMSOL recognizes there is a logical coupling inherent in the model design. When it is added as part of a predefined coupling, the coupling features are also included under the Multiphysics node. Otherwise, all relevant coupling features are available from the context menu. Additional functionality is also based on the add-on modules.

Predefined multiphysics interfaces provide you with a quick entry point for common multiphysics applications. You can create the same couplings using any of the other methods for multiphysics modeling, and you can continue to add, modify, disable, and remove physics in a model when you start using one of the predefined multiphysics interfaces. If instead you decide to add physics sequentially, this approach is also useful as you can verify that each type of physics or equation gives the expected results before adding more complexity to the model by adding another physics or coupling fields.

For this version of COMSOL Multiphysics, and depending on the add-on module, some multiphysics interfaces are not yet converted to using a predefined multiphysics coupling approach.

For links to more information about the add-on modules and the multiphysics interfaces available go to www.comsol.com/comsol-multiphysics.

Uncoupling a Multiphysics Coupling



Electromagnetic Heat Source

Electromagnetic:

Electric Currents {ec}

Heat transfer:

Heat Transfer in Solids {ht}

Electromagnetic Heat Source

Electromagnetic:

Electric Currents {ec}

Heat transfer:

None

Figure 3-13: Uncoupling a predefined multiphysics coupling feature.

For each multiphysics coupling feature, for example the Electromagnetic Heat Source, there is a section that defines the physics involved in the multiphysics coupling. By default, the applicable physics interface is selected in the lists to establish the coupling.

You can also select **None** from the lists to uncouple the node from a physics interface. If the physics interface is removed from the **Model Builder**, for example if a **Heat Transfer in Solids** physics interface is deleted, then the **Heat transfer** list for the **Electromagnetic Heat Source** reverts to **None** (Figure 3-13) as there is no interface to couple to.

If a physics interface is deleted and then added to the model again, and in order to re-establish the coupling, you need to choose the physics interface again from the lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface.

Model Inputs and Multiphysics Couplings

Model inputs can appear in an equation node's or material model node's **Model Inputs** section. Model inputs are fields such as temperature and velocities that act as inputs for material models and model equations. They appear in the **Model Inputs** section if a material is defined so that a material property becomes a function of the temperature, for example. COMSOL connects the model input to an existing field (dependent variable) within the physics interface (but not to available fields in other physics).

With more than one physics in the model, coupling of the fields is easy: all applicable fields that can serve as inputs in another physics automatically appear in the other physics interface's settings window's **Model Inputs** section. For example, with a Heat Transfer in Fluids (ht) interface and a Laminar Flow (spf) interface, you can select **Velocity field (spf/fp1)**, which the **Fluid Properties 1 {fp1}** node in the **Laminar Flow** branch defines, from the **Velocity field** list in the **Model Inputs** section of the **Heat Transfer in Fluids** node's settings window. For a coupling in the other direction (if you use temperature-dependent fluid properties, for example), you can select **Temperature (ht/fluid1)**, which the **Heat Transfer in Fluids 1 {fluid1}** node in the **Heat Transfer** branch defines, from the **Temperature** list in the **Model Inputs** section of the **Fluid Properties** node's settings window.

A list in the **Model Inputs** section becomes unavailable if the physics itself defines the field because it is then automatically connected to that field. For example, with a Heat Transfer in Fluids (ht) interface the **Temperature** list is unavailable in the **Heat Transfer in Fluids 1 {fluid1}** node. This automatic connection selects the **Temperature (ht/fluid1)** field because it is defined by the Heat Transfer in Fluids. As long as the list is unavailable, you cannot change it. If you want to use another temperature field or an expression, you first make the list editable by clicking the **Make all model inputs editable** button ( ).

In the case that you want to use another expression for these model inputs, select **User defined** to enter a user-defined value or expression in the associated text field.

See Marangoni Convection for an example of combining the Laminar Flow and Heat Transfer in Fluids interfaces (model library path **COMSOL_Multiphysics/Multiphysics/marangoni_convection**).

Specifying Model Equation Settings

The fundamental mathematical model, representing the physics in a physics interface, is contained in physics nodes with selection on the same space dimension as the physics itself. The first node under a physics branch is of this type and sets up default equations where the physics is active. These equations are controlled by specifying:

- Material properties, which COMSOL uses as coefficients in the equations
- A coordinate system, in which you can specify anisotropic material properties
- A material model, which selects an equation suitable for a given type of material

Not all physics allow anisotropic materials or more than one material model. Therefore, these settings cannot be present.

The default node uses the same material model, and thus the same equations, everywhere. Material properties can vary between different parts of the feature's selection, if the property is specified as taken **From material**. Add additional nodes to use different material models for different parts of the geometry, or to use different **User defined** material property values.

In equation-based modeling, provided by the Mathematics branch interfaces, the form of the equation is fixed for each particular node type. Each given equation form contains a number of free PDE coefficients, which you can be specify in the settings to define the specific equation that you want to solve.

Equation-Based Modeling

Specifying Equation Coefficients and Material Properties

To specify an equation coefficient or a material property, enter a value or an expression directly in the corresponding field. Such expressions can contain:

- Numerical values.
- Units (see Using Units).
- Built-in Mathematical and Numerical Constants.
- Spatial coordinates, time, and the dependent variables in any physics in the model as well as their spatial derivatives and time derivatives.
- Physical Constants—built-in universal physical constants.
- User-defined parameters, variables, coupling operators, and functions, including external functions and MATLAB functions (requires the COMSOL LiveLink™ *for* MATLAB®). See Operators, Functions, and Constants.
- Built-in functions and operators such as `d` and `mean`.

You can use these types of variables, constants, functions, and operators in all settings for the physics interfaces; many types of variables are also available anywhere in the model.

Modeling Anisotropic Materials

Anisotropic materials respond differently to an excitation depending on its direction. Because excitations are generally vectors and the corresponding response is a vector density, material properties are usually rank-2 tensor densities. For example, the following material properties are anisotropic tensor densities: diffusion coefficient, permittivity, thermal conductivity, and electrical conductivity.

These properties are, in principle, specified in matrix form and defined by their components in the coordinate system selected in the node settings. At most four components are used in 2D and at most nine components in 3D. When the material contains symmetries, you can specify only a few coefficients which are expanded to a matrix using the following patterns:

- **Isotropic** (the default)—enter only one value c.

$$C = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & c \end{bmatrix}$$

- **Diagonal**—enter the diagonal components for an anisotropic material with the main axes aligned with the model's coordinate system.

$$C = \begin{bmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & 0 & c_{33} \end{bmatrix}$$

- **Symmetric**—enter a symmetric matrix using the diagonal components and the upper off-diagonal components.

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}$$

- **Anisotropic**—enter the full 2-by-2 (2D) or 3-by-3 (3D) matrix for an anisotropic material:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Specifying Initial Values

An **Initial Values** node is added by default to each physics interface.

In some types of analyses initial values must be provided:

- As the initial condition for a time-dependent analysis.
- As an initial guess for the nonlinear stationary solver.
- As a linearization (equilibrium) point when solving a linearized stationary model or when performing an eigenvalue study.

To enter initial values, in the **Model Builder**, click the **Initial Values** node under a physics interface node. In the settings window, enter the **Initial Values** for all dependent variables (fields) in the physics. The default initial values are usually zero.

For some physics you can also enter initial values for the first time derivative of the dependent variables. These are used when solving time-dependent problems containing second time derivatives (wave-type applications). Like other default settings, these initial values apply to all domains where no other values are specified.

To use different initial values in different domains, right-click the main physics node and select **Initial Values** to add additional nodes to the Model Builder.

See Dependent Variables for more information about handling and plotting initial values.

Equation View

Equation View ($\frac{\partial u}{\partial t}$=f) is a subnode available for all the physics nodes. To display these subnodes, click the **Show** button (👁) and select **Equation View** from the **Model Builder**.

The **Equation View** settings window contains detailed information about the implementation of each physics feature: variables, shape functions, weak-form equation expressions, and constraints.

To update the values in the **Equation View** node's settings window to reflect the latest changes in a physics feature, click the **Refresh equations** button (↻) in the settings window's toolbar.

⚠	Editing the predefined expressions for variables, equations, and constraints means that the equations are altered and that COMSOL solves the model using the new expressions.

You can edit the values of variables, weak-form expressions, and constraints in the corresponding tables. This makes it possible to introduce custom changes to the equations and variable definitions. If the expression that defines a variable, for example, does not fit inside of the text field, a tooltip displays the entire expression.

For a changed definition of a variable or a change to a weak-form expression or constraint, a warning icon (⚠) appears in the leftmost column, and a small padlock is added to the lower-right corner of the icon for the physics node where you have made modifications in its equation view. To restore only the change in the selected variable, weak-form expression, or constraint, click the **Reset selected** button (↩) under the table in the **Variables**, **Weak Expression**, or **Constraints** section. To reset all changes in the equation view, click the **Reset all** button (↩) in the settings window's toolbar. If no changes remain, the padlock disappears from the corresponding physics node. An orange color for the expression that defines the variable is a warning that the unit of the expression does not match the expected unit for the variable that it defines.

🔍	For information about the Equation displays available, see Physics Nodes—Equation Section.

VARIABLES

This section has a table with the variables that the physics node defines. The table includes these columns:

- **Name**: the name of the variable.
- **Expression**: the expression, using COMSOL syntax, that defines the variable.
- **Unit**: the unit for the variable (in the active unit system). If the unit of the expression does not match the unit of the variable, the expression is displayed in orange.
- **Description**: a description of the variable.

- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the variable is defined (**Domain 1**, for example).
- **Details**: this column contains some details about the variable's behavior. See About the Details Column below.

SHAPE FUNCTIONS

This section has a table with the dependent variables that the physics node defines and their shape functions. This is primarily applicable to equation model nodes; for most physics nodes such as boundary conditions, the table is empty. The table has these columns:

- **Name**: the name of the variable.
- **Shape function**: the type of shape function (element) for the variable (for example, **Lagrange** for Lagrange elements, which are the most common elements).

Selecting an Element Type

- **Unit**: the unit for the variable (in the active unit system).
- **Description**: a description of the variable.
- **Shape frame**: the frame type (typically either a spatial or a material frame) for the shape function.
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the shape function is defined (**Domain 1**, for example)
- **Details**: This column contains some details about the shape function's behavior. See About the Details Column below.

WEAK EXPRESSIONS

This section has a table with the weak-formulation equation contributions that the physics feature generates. The table consist of the following columns:

- **Weak expression**, the equation expressed in a weak formulation
- **Integration frame**, the frame type (typically either a spatial or a material frame) used when integrating the expression.
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the weak expression is defined (**Domain 1**, for example)

Each equation contribution appears on its own row under **Weak expression**, but the order is not significant.

The PDE interfaces and the ODEs and DAEs interfaces do not display any weak expressions. They are either implemented using strong formulations, directly display the weak formulation, or define equations discretized in the time domain only.

CONSTRAINTS

This section has a table with the constraints that the physics node generates. This is typically the case for boundary conditions of constraint types, such as prescribed displacements, temperature, or velocities. Many other physics nodes do not generate any constraints, and the table is then empty. The table consists of the following columns:

- **Constraint**: the expression for the constraint

- **Constraint force**: the expression that defines the associated constraint force, which is typically the test function of the constraint
- **Shape function**: the type of shape function (element) for the constraint (for example, **Lagrange** for Lagrange elements)
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the constraint is defined (**Boundaries 1–5**, for example)

ABOUT THE DETAILS COLUMN

The **Details** column shows some details about the behavior of variables and shape functions. For variables:

- An empty cell indicates that overlapping contributions are overridden.
- **+ operation** indicates that overlapping contributions are added.
- For some variables, **Meta** indicates that the variable definitions are fully updated when solving the model. It is therefore not recommended to edit the expressions for such variables.
- In rare cases, other operations (*** operation**, for example) can occur.

For shape functions:

- **Slit** means that the shape function creates a slit for the degree of freedom

Physics Nodes—Equation Section

For each physics node there is an **Equation** section always available on the settings window. This has options to display mathematical equations applicable to the node.

Equation View

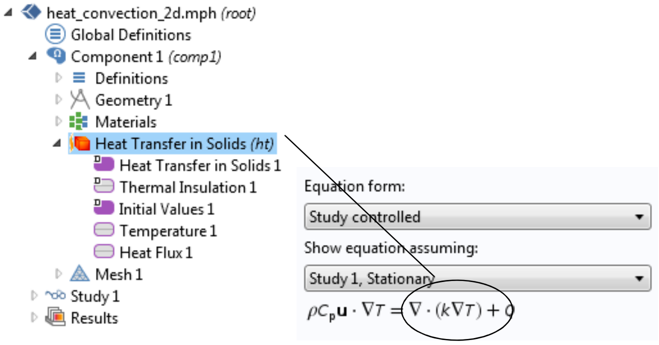
You can expand the section at any time from the settings window. Click the **Expand Sections** button (≣) in the **Model Builder** and select **Equations**.

The display options available from the lists depend on the study types and other physics-specific factors. See Figure 3-15 for an example comparing the equations that display for a **Stationary** or **Time Dependent** study for a **Heat Transfer in Solids** interface. Some settings windows do not have any options and only display the relevant equation and other windows have additional sections that become available for the **Equation** display based on the study type selected.

Study Types

Node Contributions Display a Dotted Line Under Part of the Equation

For all physics nodes (excluding the main physics interface node level), the equation that displays includes a dotted line underneath where the node's contribution is made to the equation. See Figure 3-14 for an example where a section of the heat transfer equation is underlined, indicating where the **Heat Transfer in Solids** node contributes to it.



heat_convection_2d.mph *(root)*
Global Definitions
Component 1 *(comp1)*
Definitions
Geometry 1
Materials
Heat Transfer in Solids *(ht)*
Heat Transfer in Solids 1
Thermal Insulation 1
Initial Values 1
Temperature 1
Heat Flux 1
Mesh 1
Study 1
Results

Equation form:
Study controlled

Show equation assuming:
Study 1, Stationary

$\rho C_p \mathbf{u}\cdot\nabla T = \nabla\cdot(k\nabla T) + Q$

Figure 3-14: The Heat Transfer in Solids contribution to the equation for a 3D model.

Equation Form

When you add physics to a Component, the default **Study types** are listed in the **Equation form** list. **Study controlled** is the default; select another option as required.

Show Equation Assuming

The **Show equation assuming** option is available by default when **Study controlled** is selected (or left as the default) as the **Equation form**. Availability of the options are based on the studies added and defined for the model.

For the following options—frequency and mode analysis frequency—you also have the option to use another frequency than the one used by the solver. This can be necessary if you need two different frequencies for two physics.

Frequency

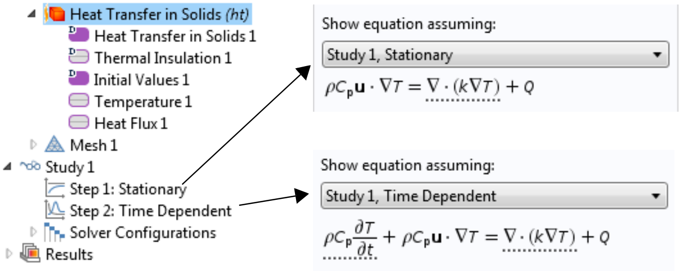
This option is available if **Frequency domain** is selected as the **Equation form**. The default uses the frequency **From solver**. If **User defined** is selected, enter another value or expression (SI unit: Hz).

Mode Analysis Frequency

This option is available if **Mode Analysis** or **Boundary Mode Analysis** is selected as the **Equation Form**. Enter a value or expression in the field (SI unit: Hz). Specify a frequency (it is not present as a solver variable).

Port Name

This option is available with the RF Module **Electromagnetic Waves** interface and if **Boundary Mode Analysis** is selected as the **Equation Form**. Enter a value in the field (unitless).



Heat Transfer in Solids *(ht)*
Heat Transfer in Solids 1
Thermal Insulation 1
Initial Values 1
Temperature 1
Heat Flux 1
Mesh 1
Study 1
Step 1: Stationary
Step 2: Time Dependent
Solver Configurations
Results

Show equation assuming:
Study 1, Stationary

$\rho C_p \mathbf{u}\cdot\nabla T = \nabla\cdot(k\nabla T) + Q$

Show equation assuming:
Study 1, Time Dependent

$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u}\cdot\nabla T = \nabla\cdot(k\nabla T) + Q$

Figure 3-15: An example of the Equation section on a Heat Transfer interface. Selecting the study type updates the equation accordingly.

Boundary Conditions

In the interior of the selection where a physics is active, its behavior is governed by its model equations and material properties. *Boundary conditions* apply to the geometric entities separating this region from the unspecified outside, and also to interior entities of the same dimension. Therefore, boundary conditions on a 3D solid object apply to the exterior surfaces of the solid, and to interior surfaces embedded in the solid. On a shell geometry in 3D space, boundary conditions instead apply to the edges of the shell surface. In general, boundary conditions apply to geometric entities whose dimension is one less than the physics interface's dimension.

All physics branches that contain a default model equation node also contain a default boundary condition node. This boundary condition is active on all exterior boundaries of the physics' selection, except on the symmetry axis of axisymmetric 2D models. On interior boundaries, an implicit continuity condition applies, which makes the physics field (the temperature, for example) continuous across interior boundaries.

Boundary Condition Types

There are two fundamental ways to specify what is happening at a boundary, and two corresponding fundamental boundary condition types:

- *Flux conditions* specify how the surroundings affect and interact with the model at the boundary, often expressed as an applied force, flux, or current. This type of boundary condition is also called a *Neumann boundary condition*.
- *Constraints* specify the result of the interaction between the model and its surroundings, expressed as expected values of the dependent variables. This type of boundary condition is also called a *Dirichlet boundary condition*.

The two types are closely related because in a well-posed model, every flux condition results in some unique values of the dependent variables, and every constraint requires a unique flux to enforce the expected values. Which type of condition to use depends on what is known about the conditions at the boundary: if the flux is known, the model computes the dependent variables for you; if the values of the dependent variables are known, the model computes the flux.

FLUX CONDITIONS

Flux boundary conditions specify the component of a vector or tensor quantity in the direction normal to the boundary, per unit area of the boundary. Typical examples of flux conditions are the specification of:

- A Boundary Load in a solid model, which prescribes the stress acting on the boundary.
- Heat Flux in a heat transfer model, which prescribes the heat per unit area flowing into (or out of) the model across the boundary.
- A Normal Current Density in an AC/DC model, which prescribes the electrical current per unit area entering (or exiting) the model at the boundary.

There are also more advanced types of flux conditions, where the flux or force is calculated based on local values of dependent variables and other parameters. For example, a **Convective Heat Flux** boundary condition on a heated body computes the heat flux based on a heat transfer coefficient and the temperature difference to the surroundings.

Convective Heat Flux requires the CFD Module or Heat Transfer Module.

In COMSOL, by convention, the force acting *on* the model or the flux *into* the model is specified. That is, specify how the surroundings affect the model and not how the model affects its surroundings.

CONSTRAINTS

Constraint boundary conditions specify the value of one or more dependent variables at the boundary, or a relationship between two or more dependent variables. Typical examples include specifying:

- A Prescribed Displacement of the boundary of a solid object.
- That the velocity is zero on a Wall boundary in a CFD model.
- The Temperature at the boundary of a heated solid.
- The Electric Potential on an electrode in an AC/DC model.

Examples specifying a relation between dependent variables include Roller conditions on solids and Wall conditions for slip flow.

Because constraint conditions generally specify the value of a dependent variable, they also provide a reference level for that variable—which a flux condition normally does not. In many types of physics, the model equations together with only flux boundary conditions uniquely describe the local behavior of the dependent variable, but leave the global level undefined. From a physical point of view, the absolute value of the dependent variables are often of less interest, but the existence of a single, unique, solution is essential for some solvers.

Therefore it is often necessary to apply at least one constraint condition in a model, to provide a global reference value for the dependent variables. For example, it is common to designate one of the electrodes in an AC/DC model as Ground, which constrains the electric potential there to zero and gives a reference with which to compare other parts of the model.

In most physics, the default boundary condition is of *flux* type and does *not* fix a reference level for the dependent variable. Therefore when solving certain study types, notably **Stationary** studies, you must manually add at least one boundary condition of *constraint* type (or a point constraint) for the model to be well-defined.

SWITCHING OFF A CONSTRAINT

If you want to model a constraint that is active only for a certain period of time in a time-dependent simulation, for example, you can use the fact that a 0 constraint (or a Dirichlet boundary condition $u = u$) means that there is no constraint; instead, the boundary condition becomes a “no flux” or “insulation” condition. To implement such a time-limited constraint you can use the `if` operator: for example, for a Dirichlet boundary condition, `if(t<2,1,u)` means that for $t < 2$, u is equal to 1 but at $t = 2$ the boundary condition is turned off by setting $u = u$. For a Constraint node, the corresponding `if` statement is `if(t<2,1-u,0)`.,

- Constraint Reaction Terms
- Weak Constraints
- Constraint Settings

Physics Boundary Types

There are different types of boundaries for the physics, which all support different types of boundary conditions:

- Exterior boundaries, where most boundary conditions are applicable—see below.

- Interior boundaries, where special interface conditions can be applicable—see below.
- Axial symmetry boundaries, which are artificial boundaries representing the symmetry axis in axisymmetric models.

If a selection for a boundary condition node, for example, contains boundaries of a type that is not applicable or supported, the **Selection** list has **(not applicable)** next to those boundary numbers.

INTERIOR AND EXTERIOR BOUNDARIES

When specifying boundary and interface conditions, COMSOL differentiates between exterior and interior boundaries:

- An *exterior boundary* is an outer boundary of the modeling domain.
- A *interior boundary* is a dividing interface between two domains in the geometry.

If an equation or physics interface is deactivated in one domain, the interior boundary between the active and inactive domain becomes an exterior boundary for its variables because it then borders on the outside of the active domain for those fields. The boundaries of the inactive domain are then void.

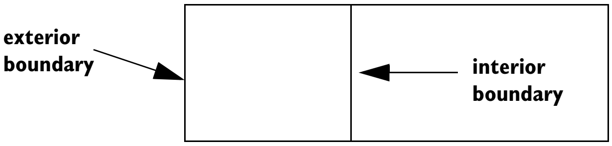


Figure 3-16: Examples of exterior and interior boundaries.

Continuity on Interior Boundaries

Unless a boundary condition is specified on interior boundaries (such as a contact resistance condition), COMSOL ensures continuity in the field variables across interior boundaries. For assembly geometries with identity pairs, select a **Continuity** node on the **Pairs** menu in the boundary part of the context menu for most physics. The **Continuity** condition is only suitable for pairs where the boundaries match.

BOUNDARY SELECTION

The selection list in this section shows the boundaries for the selected pairs.

PAIR SELECTION

Select the pairs where you want to impose continuity across the pair boundaries. Select the pairs from the **Pairs** list (Ctrl-click to deselect).

Identity and Contact Pairs

Physics Axial Symmetry Node

In axisymmetric models, boundaries on the symmetry axis are boundaries where only a condition for the axial symmetry exists. COMSOL adds a default **Axial Symmetry** node that is active on all boundaries on the symmetry axis. The condition on the symmetry axis is typically a zero Neumann or no-flux condition.

BOUNDARY SELECTION

The selection list for boundaries is not available because this is a default boundary condition. In the list, boundaries that are not on the symmetry axis have **(not applicable)** added after the boundary number.

Physics Nodes by Space Dimension

Constraint Reaction Terms

Enforcing a constraint condition is more or less a matter of finding a corresponding flux condition that leads to the desired values of the dependent variables. The hidden flux conditions introduced this way appear as *reaction terms* in the system of equations modeling the physics. These reaction terms normally have a physical meaning and correspond to a flux condition, for example:

- The reaction term enforcing a Prescribed Displacement on a solid model is a *reaction force*, similar to a Boundary Load boundary condition.
- The reaction term enforcing a **Pressure** in an acoustics model is a Normal Acceleration.
- The reaction term enforcing a **Temperature** in a heat transfer model is a Heat Flux.

The reaction terms in the model equations can be scaled in different ways, affecting mainly the numerics and solvers. In a model with more than one dependent variable, it is also possible to distribute the reaction fluxes, or forces, over the variables in different ways—while still enforcing the original constraint.

SYMMETRIC REACTION TERMS

Most boundary conditions of constraint type, by default, introduce reaction terms in such a way that an otherwise symmetric system of equations remains symmetric. This makes constraints *bidirectional* in the sense that all dependent variables that appear in a constraint expression are also affected by the reaction terms.

To illustrate this, suppose a **Prescribed Displacement** boundary condition is applied on a solid model, specifying that the x-displacement of the boundary, u, is proportional to the y-displacement, v, with a constant of proportionality, k, which is a function of the boundary temperature T:

$$u = k(T)v \tag{3-1}$$

If fully symmetric reaction terms are used to enforce this constraint, reaction forces are applied on both displacement components u and v, as well as a reaction heat flux in the heat transfer equation. Applying symmetric reaction terms this way, on completely different equations, usually makes no sense.

In particular, the solid displacement equation and the heat transfer equation have different units. Because you can choose length and temperature unit independently, the relative scale of the equations is undefined and the symmetry of the coupled system irrelevant. Further you would not, from a physical point of view, expect a constraint on the displacement of a solid boundary to directly affect the temperature field in a model.

RESTRICTED AND NONSYMMETRIC REACTION TERMS

As an alternative to the default (symmetric) application of reaction terms, you can choose to have these affect only the equations and variables in the physics interface where the constraint boundary condition is added. For the example in Equation 3-1, the reaction terms can be restricted to act on the displacement variables and equations in the Solid Mechanics interface, leaving the temperature unaffected. Many different restrictions of this type are possible, in principle, and COMSOL generally provides two alternatives:

- The most consistent and general way to avoid spurious reaction terms affecting other physics is to start from the globally symmetric formulation and remove the terms entering equations belonging to other physics interfaces.

This limits the reaction terms to affecting the current physics in such a way that if there are no other physics in the model, so the reaction terms preserve the symmetry. For Equation 3-1, this means that reaction terms are distributed over both u and v equations, in proportions $1{:}k(T)$.

- The other alternative is to apply the reaction terms only on certain individual variables. Another way to look at Equation 3-1is to read it as prescribing a value for the x-displacement u, rather than prescribing a given relation between u and v. Accepting that view, it is reasonable to insert reaction terms only acting on u. Such reaction terms, in general, do not preserve symmetry even for a single physics interface.

Weak Constraints

The standard method to enforce constraints in COMSOL applies the constraints pointwise at node points in the mesh. At each node point, only local values of the dependent variables are affected by the constraint, making the constraints independent of each other. The solvers can therefore eliminate both the constrained degrees of freedom and the constraint force terms, effectively reducing the system of equations and decreasing the number of degrees of freedom being solved for.

Weak constraints enforce the constraint in a local average sense, using shape functions as weights. Reaction terms are explicitly included in the system of equations, which is extended with Lagrange multiplier variables. These Lagrange multipliers in general have a physical meaning and an interpretation as a constraint force or flux. Whereas a standard constraint decreases the number of degrees of freedom by the number of unique constraints, weak constraints increase the degrees of freedom by the same number.

A weak constraint is respected only on average over each Lagrange multiplier shape function, rather than pointwise at mesh nodes. When it is possible to satisfy the constraint everywhere on each mesh element, standard and weak constraints in general lead to the same solution. Conversely, when constraints are contradictory or impossible to satisfy everywhere, standard and weak constraints can distribute the error differently, and therefore lead to slightly different solutions.

Weak constraints can be of use in the following situations:

- Standard constraints must never contain time derivatives of the dependent variables. Weak constraints do not have this limitation because they allow the same variables as any other term in the combined system of equations. Note that reaction terms cannot be applied symmetrically to time derivatives but must be selectively applied to individual variables.
- When the reaction force or flux is needed during a solution, because it enters into a coefficient somewhere, the Lagrange multiplier from a weak constraint can provide an accurate value (see Computing Accurate Fluxes). The corresponding variables computed from derivatives of the dependent variables are not as accurate and can, if used, introduce considerable errors in the solution.
- When constraints are strongly nonlinear, weak constraints can allow faster and more robust convergence. For nonlinear constraints, the true linearized subproblem solved in each solution step depends on the value of the Lagrange multiplier variables from the previous step. When using standard constraints, this information is discarded between solution steps. Using weak constraints, the Lagrange multiplier values are instead retained between steps because they are part of the solution vector.

Compared to standard, eliminated, constraints, weak constraints can also have the following drawbacks:

- Discontinuous constraints result in (theoretically) infinite Lagrange multipliers. In practice, large oscillations result.
- Pointwise and weak constraints on the same set of variables on adjacent boundaries (that is, boundaries that share common node points in the mesh) do not work. This means that if all boundaries must be constrained on a solid and you want to use a weak constraint on one boundary segment (one face), the weak constraint must be used on the entire boundary of the solid (if the boundary is connected).

- Lagrange multipliers are in some cases difficult to interpret. For example, Lagrange multipliers from Dirichlet conditions in axial symmetry are not equal to the reaction flux per area but rather per length and full revolution. For separate Weak Constraints nodes in axial symmetry, the default quadrature settings include a multiplication by $2\pi r$, making the Lagrange multiplier represent flux per area.
- Because extra unknowns are introduced for the Lagrange multipliers, the size of the problem increases compared to the standard constraint elimination method.
- The Lagrange multiplier variables added by the weak constraints have a different unit than the main system variables and can therefore be of a completely different order of magnitude. This can lead to scaling problems. Usually the automatic variable scaling in the solvers is sufficient, but there are cases when manual scaling is needed.
- Weak constraints introduce zeros on the main diagonal of the Jacobian matrix of the discretized system, which therefore cannot be positive definite. This makes certain linear solvers and preconditioners unavailable for solving problems with weak constraints. In particular, the conjugate gradients iterative solver does not work, and neither does the SOR class of preconditioners and smoothers. Instead, try another iterative solver and use the Vanka algorithm with the Lagrange multipliers as the Vanka variables, or use the incomplete LU factorization algorithm as preconditioner.

Constraint Settings

Most constraint nodes have a **Constraint Settings** section which is only available when **Advanced Physics Options** is selected from the **Show** menu ( ). This section provides settings controlling how reaction terms are applied and whether standard or weak constraints are used. Choose to **Apply reaction terms on**:

- **All physics (symmetric)** to apply reaction terms symmetrically on all dependent variables taking part in the constraint.
- **Current physics (internally symmetric)** to apply reaction terms symmetrically only on the dependent variables in the physics where the constraint is added. This leaves other physics unaffected by the constraint.
- **Individual dependent variables** to apply reaction terms only on selected variables. For most physics, this makes the constraint *unidirectional* and often nonsymmetric.

Select the **Use weak constraints** check box to replace the point-wise standard constraints with weak constraints. Note that this introduces additional equations and dependent variables.

> Not all constraints provide all the above options. Some reaction term methods can be missing and weak constraints are not allowed. Some constraint nodes can also implement additional options.

Periodic Boundary Conditions

Use *periodic boundary conditions* to make the solution equal on two different (but usually equally shaped) boundaries.

To add a periodic boundary condition, in the **Model Builder**, right-click a physics interface node and select **Periodic Condition**. The periodic boundary condition typically implements standard periodicity so that $u(x_0) = u(x_1)$ (that is, the value of the solution is the same on the periodic boundaries), but in most cases you can also choose antiperiodicity so that the solutions have opposing signs so that $u(x_0) = -u(x_1)$. For fluid flow physics interfaces, the **Periodic Flow Condition** provides a similar periodic boundary condition but without a selection of periodicity. Typically, the periodic boundary conditions determine the source and destination boundaries automatically (and display them, under **Component>Definitions**, in an **Explicit** selection node ( ), which is "read only"), but you can also define feasible destination boundaries manually by adding a **Destination Selection** subnode.

For some physics you can choose the direction in which you want a periodic boundary condition. For a description of the standard periodic boundary condition, which most physics use, see Periodic Condition.

The KdV Equation and Solitons: model library path **COMSOL_Multiphysics/Equation-Based_Models/kdv_equation**.

PERIODIC BOUNDARY CONDITION MODEL EXAMPLES

In addition to the KdV Equation model example, other modules have examples using this feature:

AC/DC Module

Magnetotellurics: model library path: **ACDC_Module/Other_Industrial_Applications/magnetotellurics**

Acoustics Module

Porous Absorber: model library path: **Acoustics_Module/Industrial_Models/porous_absorber**

RF Module or Wave Optics Module

Fresnel Equations: model library path: **RF Module** or **Wave Optics Module/Verification_Models/fresnel_equations**

Structural Mechanics Module

Vibrations of an Impeller: model library path: **Structural_Mechanics_Module/Dynamics_and_Vibration/impeller**

Computing Accurate Fluxes

Flux Computation Methods

COMSOL provides three ways of computing accurate fluxes and reaction forces:

- The first approach involves the reaction force operator (`reacf`) that makes it possible to compute integrals of reaction forces or fluxes during analysis. See reacf for details.
- The second, more general approach for calculating reaction forces and fluxes is to use weak constraints. Use this approach when you need reaction forces or fluxes in other contexts than calculating integrals of reaction forces or fluxes.

See Weak Constraints for more information about using weak constraints.

- Some physics provide a third way of computing accurate fluxes. Under the **Discretization** section, select the **Compute boundary fluxes** check box. The solver then computes variables storing an accurate boundary flux from each boundary into the adjacent domain (in addition to the standard extrapolated value). On interior boundaries, there are two flux variables corresponding to the flux into the domains on either side of the boundary. Unlike the other methods, these variables are available also on unconstrained boundaries. This method is active by default in Coefficient Form PDE, General Form PDE, heat transfer, and mass transport interfaces. There is also an **Apply smoothing to boundary fluxes** check box that is selected by default. The smoothing can provide a more well-behaved flux value close to singularities.

When using weak constraints in interfaces, the Lagrange multipliers are additional dependent variables in those physics interfaces. When using the reaction force operator, the reaction force operator of a certain dependent variable corresponds to the Lagrange multiplier of that dependent variable. The Lagrange multipliers correspond to the following quantities in the physics interfaces:

TABLE 3-5: INTERPRETATION OF LAGRANGE MULTIPLIERS

PHYSICS	QUANTITY
Electrostatics	Surface charge density
Magnetic Fields	Surface current
Electric Currents	Current density
Heat Transfer	Heat flux
Transport of Diluted Species	Flux
Solid Mechanics	Force per area
Pressure Acoustics	Normal displacement (acceleration for eigenfrequency studies)
Laminar Flow	Total force per area

The sign of the Lagrange multiplier is the same as the one used when applying the corresponding quantity explicitly in a flux condition. As a general rule, the sign corresponds to an action by the surroundings on the model, rather than the opposite.

COMSOL computes only the part of the boundary flux captured by the Lagrange multiplier. You might have additional flux coming from boundary sources or nonidentity constraint matrices. This should not happen in the physics interfaces, though.

Lagrange multipliers in axial symmetry are not equal to the reaction flux per area but rather per length and full revolution.

Flux Calculation Example—Heat Transfer Model

The reaction forces are computed from the value of the residual vector L at every node point where a constraint is applied. Therefore, the reaction forces should be thought of as discrete values at each node point rather than continuous fields.

The boundary flux variables are computed in a similar way to the reaction forces but with two important differences:

- First, on each boundary, the contributions to the residual vector from the boundary and from the adjacent domains are computed separately. This makes it possible to compute the flux into each adjacent domain even when there is no constraint on the boundary so that the full residual vector is zero.
- Second, the nodal fluxes computed from the residual vector are further processed and represented as a continuous field on the boundary. The integral of this flux field over a boundary is equal to the sum of the nodal fluxes.

Flux Calculation Example—Heat Transfer Model

Consider a heat transfer model where a heat flux of 1 W/m^2 flows in through one boundary of a square 2D region. All other boundaries are kept at a fixed temperature of 293.15 K. The material is copper. This example verifies that the flux is conserved exactly using a Lagrange multiplier for computing the total flux over the boundaries with a fixed temperature.

MODEL WIZARD

1 On the **Select Space Dimension** page, click the **2D** button.

2 In the list of physics, select open **Heat Transfer>Heat Transfer in Solids**. Then click the **Add** button.

3 Click the **Study** button. Then on the **Select Study** page, select **Preset Studies>Stationary**.

4 Click **Done**.

GEOMETRY MODELING

Draw a unit square (1-by-1 m).

MATERIALS

In the **Material Browser**, locate **Built-in>Copper** and then right-click it and select **Add to Component 1**. Then click **Close**.

HEAT TRANSFER

The **Heat Transfer in Solids** node defines the material properties to be those from the material (copper) and does not need to be changed, but the default boundary condition is thermal insulation. Instead, add a heat flux to the bottom boundary and a fixed temperature on the other three boundaries.

1 In the **Model Builder** window, right-click **Heat Transfer** node and select **Heat Flux**.

2 Click boundary 2 (the bottom boundary) to add it to the selection.

3 In the **Heat Flux** node's settings window, enter `1` (1 W/m^2) in the **General inward heat flux** field for q_0.

4 In the **Model Builder** window, right-click **Heat Transfer** node and select **Temperature**.

5 Select the other three boundaries (1, 3, and 4) and add them to the selection for the temperature condition.

6 The following step is only needed to show how to use a Lagrange multiplier for an accurate flux. Built-in variables for accurate fluxes are available directly also without this step. To display the weak constraint option to add the Lagrange multipliers, click the **Show** button ( ) and select **Advanced Physics Options**. In the **Temperature** node's settings window, keep the default value for the temperature, 293.15 K, but select the **Use weak constraints** check box in the **Constraint Settings** section. This adds a Lagrange multiplier for the heat flux as an extra variable to compute.

COMPUTING THE SOLUTION

In the **Model Builder** window, right-click **Study 1** and choose **Compute**. The resulting plot shows the temperature distribution in the domain.

RESULTS—FLUX EXPRESSION AND LAGRANGE MULTIPLIER

1 In the **Model Builder** window, choose **Results>Derived Values>Integration>Line Integration**.

2 Select the three boundaries with a fixed temperature (1, 3, and 4) to add them to the selection in the **Line Integration** node's settings window.

3 Click the **Replace Expression** button ( ) and select
Heat Transfer in Solids>Boundary fluxes>Normal total heat flux (the variable `ht.ntflux`).

4 Click the **Evaluate** button (=) at the top of the settings window. The total normal heat flux across these boundaries appears in the **Table** window under **Normal total heat flux (W/m)** and is exactly equal to the influx of 1 W/m (the normal flux is by convention positive in the direction of the normal). If you would clear the **Compute boundary fluxes** check box in the **Discretization** section of the Heat Transfer in Solids node's settings window, and the re-solve the model, the same flux variable is not as accurate and has a value of about 0.986 W/m. That value approaches 1 if you refine the mesh.

5 Click the **Replace Expression** button ( ) and select **Heat Transfer>Lagrange multiplier for temperature** (the variable `T_lm`).

6 Click the **Evaluate** button (=) at the top of the settings window. The total heat flux across these boundaries appears in the **Table** window under **Lagrange multiplier for temperature** and is −1—exactly equal to the influx (but with opposite sign) without the need for a computationally expensive extremely fine mesh. This makes this method useful for physics where built-in accurate flux variables are not available.

Numerical Stabilization

About Numerical Stabilization in COMSOL

This section discusses the numerical stability of the generic scalar convection-diffusion transport equation

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + F \tag{3-2}$$

where β is the convective velocity vector, c is the diffusion coefficient, u is a transported scalar, and F is a source term. The underlying finite element discretization method in COMSOL Multiphysics is the Galerkin method. When discretizing Equation 3-2 using the Galerkin method, it is well known that the resulting numerical problem becomes unstable for an element Péclet number (Pe) larger than one (Ref. 1):

$$\mathrm{Pe} = \frac{\|\beta\| h}{2c} > 1 \tag{3-3}$$

where h is the mesh element size. The Péclet number is a measure of the relative importance of the convective effects compared to the diffusive effects; a large Péclet number indicates that the convective effects dominate over the diffusive effects.

Oscillations can occur where any of the following conditions exist and the Péclet number exceeds one:

- A Dirichlet boundary condition can lead to a solution containing a steep gradient near the boundary, forming a boundary layer. If the mesh cannot resolve the boundary layer, this creates a local disturbance.
- A space-dependent initial condition that the mesh does not resolve can cause a local initial disturbance that propagates through the computational domain.
- A small initial diffusion term close to a nonconstant source term or a nonconstant Dirichlet boundary condition can result in a local disturbance.

As long as diffusion is present, there is—at least in theory—a mesh resolution beyond which the discretization is stable. This means that the spurious oscillations can be removed by refining the mesh. In practice, this method is seldom feasible because it can require a very dense mesh. Instead, it is common practice to use stabilization methods; that is, methods that add artificial diffusion. COMSOL provides several of these methods and described in An Example of Stabilization.

Consistent Stabilization and Inconsistent Stabilization Sections on Settings Windows

Numerical stabilization is available for physics interfaces that model transport such as fluid flow or convective heat transfer, where the fundamental governing equations are less stable than, for example, conduction-dominated models, solid mechanics models, and wave propagation in the frequency domain.

Several physics interfaces have these settings available, and below you find the common information about the stabilization settings. Differences not described below are noted for the individual interface documentation.

CONSISTENT STABILIZATION

To enable this section, click the **Show** button (👁) and select **Stabilization**.

There are two *consistent stabilization methods*: **Streamline diffusion** and **Crosswind diffusion**. Usually, both check boxes for these methods are selected by default and should remain selected for optimal performance. Consistent stabilization methods do not perturb the original transport equation.

The crosswind diffusion method specifies the smallest allowable concentration change across an element. As the concentration gradient appears in the denominator in the equations describing crosswind diffusion, the gradient ensures that unreasonable values do not occur in regions with small to negligible concentration changes.

Crosswind Diffusion and Tuning Parameter

In most cases when the **Crosswind diffusion** check box is selected, enter a **Tuning parameter** C_{k}. The default is most often 0.5. The **Tuning parameter** controls the amount of crosswind diffusion introduced. It recommended that it is kept in the order of 1 to avoid introducing excessive amounts of diffusion. The value used must also neither be space dependent nor time-dependent.

Crosswind Diffusion and Lower Gradient Limit

In some cases, if the **Crosswind diffusion** check box is selected, the **Lower gradient limit** g_{lim} (SI unit: K/m) field is available. This variable corresponds to the smallest concentration change across an element considered by the stabilization, and is used to make sure that the crosswind diffusion expressions are valid also in regions with small to negligible concentration changes.

Residual

In some cases, and for both consistent stabilization methods, select a **Residual** (or **Equation Residual**). **Approximate residual** is the default setting and it means that derivatives of the diffusion tensor components are neglected. This setting is usually accurate enough and is faster to compute. If required, select **Full residual** instead.

INCONSISTENT STABILIZATION

To enable this section, click the **Show** button ( ) and select **Stabilization**.

There is usually just one *inconsistent stabilization method*— **Isotropic diffusion**. This method is equivalent to adding a term to the diffusion coefficient in order to dampen the effect of oscillations by making the system somewhat less dominated by convection. If possible, minimize the use of the inconsistent stabilization method because by using it you no longer solve the original problem.

By default there is no isotropic diffusion selected. If required, select the **Isotropic diffusion** check box and enter a **Tuning parameter** δ_{id} as a scalar positive value. The default value is 0.25 (a reasonable value to start with is roughly 0.5 divided by the element order). A higher value adds more isotropic diffusion.

- An Example of Stabilization
- Stabilization Techniques

An Example of Stabilization

This example uses the Heat Transfer interface. To illustrate the concepts, consider the problem

$$\cos\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial x}+\sin\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial y} = 10^{-4}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+1 \qquad (3\text{-}4)$$

solved on the unit square. Equation 3-4 is discretized using 10 times 10 biquadratic Lagrangian elements. The boundary conditions are:

- $u = 1$ for $x = 0$
- $u = 1$ for $y = 0$
- $u = 0$ for $x = 1$
- $u = 0$ for $y = 1$

Figure 3-17 shows the mesh and boundary conditions. In general, using uniform meshes for transport problems is not recommended. Nevertheless, this example uses a uniform mesh to demonstrate the different stabilization techniques.

The expected solution rises slowly and smoothly from the left and lower boundaries and has sharp boundary layers along the upper and right boundaries. Figure 3-18 shows a reference solution obtained using 100-by-100 biquadratic Lagrangian elements with streamline diffusion and crosswind diffusion (see the next section). The arrows indicate the direction of β.

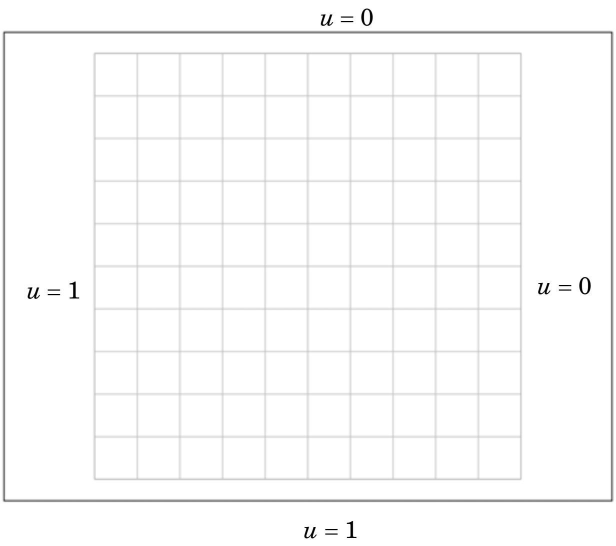


Figure 3-17: The computation domain, mesh, and boundary condition for Equation 3-4.

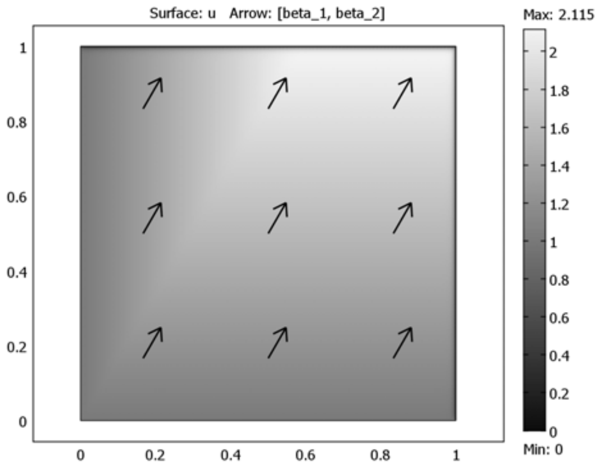


Figure 3-18: Reference solution of Equation 3-4*. Solved using 100 times 100 biquadratic elements with streamline diffusion and crosswind diffusion.*

The cell Péclet number for this example is

$$\mathrm{Pe} = \frac{1 \cdot 0.1}{2 \cdot 10^{-4}} = 500 >> 1$$

Figure 3-19 displays the solution obtained using the mesh in Figure 3-17 and (unstabilized) Galerkin discretization. As can be expected with such a high Péclet number, the unstabilized solution shows little, if any, resemblance to the reference solution in Figure 3-18. The right plot in Figure 3-19 shows a cross-sectional plot along the dashed line, $y = 0.8$ and the corresponding reference solution. Notice that the unstabilized solution is completely destroyed by oscillations.

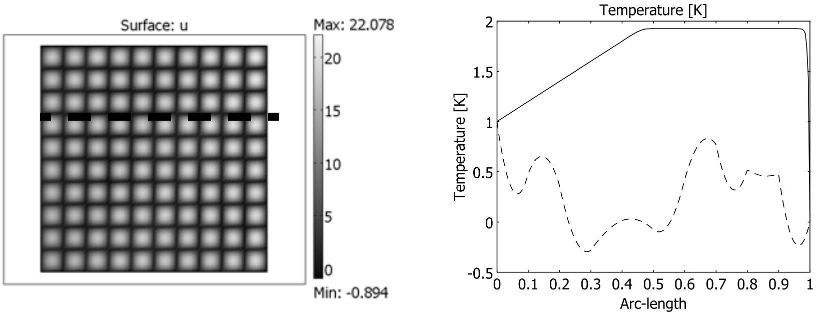


Figure 3-19: Equation 3-4 solved using unstabilized Galerkin formulation. The right plot compares the unstabilized solution (dashed line) along the dashed line in the left plot (y = 0.8) with the reference solution (solid line).

The Stabilization Techniques section explores how different stabilization techniques affect the solution of this example.

Stabilization Techniques

Several techniques for handling numerical instabilities without the need for mesh refinement are available. They all have in common that they add terms to the transport equation. These terms introduce numerical diffusion (artificial diffusion, artificial viscosity, or numerical viscosity are other common names) that stabilize the solution. To display these sections, click the **Show** button ( ) and select **Stabilization**.

- Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces
- Numerical Stability—Stabilization Techniques for Fluid Flow

CONSISTENT STABILIZATION

A consistent stabilization method adds numerical diffusion in such a way that if u is an exact solution to Equation 3-2, then it is also a solution to the problem with numerical diffusion. In other words, a consistent stabilization method gives less numerical diffusion the closer the numerical solution comes to the exact solution.

INCONSISTENT STABILIZATION

An inconsistent stabilization method adds numerical diffusion in such a way that if u is an exact solution to Equation 3-2, then it is not necessarily a solution to the problem with numerical diffusion. In other words, an inconsistent method adds a certain amount of diffusion independently of how close the numerical solution is the to exact solution.

ISOTROPIC DIFFUSION

Adding isotropic diffusion is equivalent to adding a term,

$$c_{\mathrm{art}} = \delta_{\mathrm{id}} h \|\beta\|$$

to the physical diffusion coefficient, c. Here δ_{id} is a tuning parameter. This means that you do not solve the original problem, Equation 3-2, but rather the modified $O(h)$-perturbed problem

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot ((c + c_{\mathrm{art}})\nabla u) + F \tag{3-5}$$

Hence, isotropic diffusion is an inconsistent stabilization method. If $\delta_{\mathrm{id}} = 0.5$, the new cell Péclet number can be expressed as

$$\mathrm{Pe} = \frac{h\|\beta\|}{2(c + c_{\mathrm{art}})} = \frac{h\|\beta\|}{2c + h\|\beta\|}$$

Clearly, as $\|\beta\|$ approaches infinity, Pe approaches, but never exceeds, one. While a solution obtained with isotropic diffusion might not be satisfactory in all cases, the added diffusion definitely dampens the effects of oscillations and impedes their propagation to other parts of the system. It is not always necessary to set δ_{id} as high as 0.5 to get a smooth solution, and its value should be smaller if possible. A good rule of thumb is to select $\delta = 0.5/p$, where p is the order of the basis functions. The default value is $\delta_{\mathrm{id}} = 0.25$

Figure 3-20 shows the effect of isotropic diffusion on Equation 3-4 with $\delta_{\mathrm{id}} = 0.25$. Although the solution is smooth, the comparison with the reference solution in the right plot reveals that the isotropic diffusion introduces far too much diffusion.

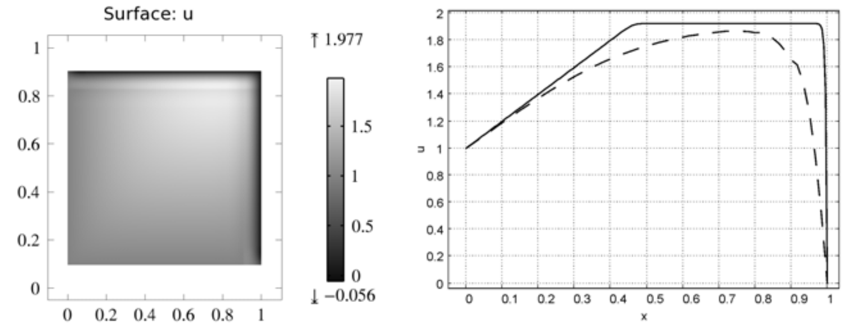


Figure 3-20: Equation 3-4 *solved using isotropic diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).*

STREAMLINE DIFFUSION

The streamline diffusion method in COMSOL is a consistent stabilization method. When applied to Equation 3-2, it recovers the streamline upwind Petrov-Galerkin (SUPG) method, but it can also recover functionality from the Galerkin least-squares (GLS) method. Both methods are described below. For theoretical details, see Ref. 1 and Ref. 2.

Streamline Upwind Petrov-Galerkin (SUPG)

The theory underlying SUPG is a bit too complicated to describe here, but the resulting expressions can be shown to be closely related to upwinding schemes in finite difference and finite volume methods. SUPG can be shown to add a smaller amount of stability than isotropic diffusion (see Ref. 3), but while the accuracy of isotropic diffusion is at best $O(h)$, the accuracy of SUPG can be shown to be at least $O(h^{p+1/2})$ where $p \geq 1$ is the order of the basis functions.

Figure 3-21 displays the effect of SUPG on the solution of Equation 3-4. The solution closely follows the reference solution away from the boundary layers, but at the boundary layers, oscillations occur. This is a typical behavior for streamline diffusion: the solution becomes smooth and exact in smooth regions but can contain local oscillations at sharp gradients.

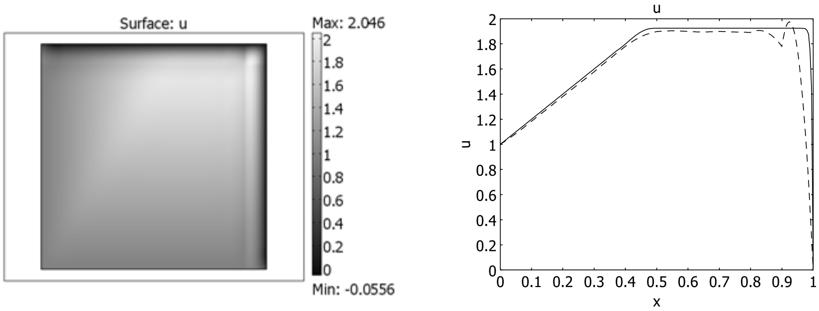


Figure 3-21: Equation 3-4 solved using streamline diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).

Galerkin Least-Squares (GLS)

Galerkin least-squares (GLS) is a more advanced version of SUPG, with which it shares many features. GLS, for example, is also a consistent method and has the same order of accuracy as SUPG. To understand the differences between GLS and SUPG, consider the following extended form of Equation 3-2:

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + su + F \tag{3-6}$$

where s is a production coefficient if $s > 0$ and an absorption coefficient if $s < 0$. If $s \neq 0$, the numerical solution of Equation 3-6 is characterized by the Péclet number (see Equation 3-3) and the element Damköhler number:

$$\mathrm{Da} = \frac{|s|h}{\|\beta\|}$$

A new dimensionless number can be formed by combining the Damköhler number and the Péclet number:

$$2\mathrm{DaPe} = \frac{|s|h^2}{c} \tag{3-7}$$

The (unstabilized) Galerkin discretization becomes unstable if $2\mathrm{DaPe} > 1$ (Ref. 4), that is, if the production/absorption effects dominate over the viscous effects. GLS differs from SUPG in that GLS relaxes this requirement while SUPG does not.[1]

CROSSWIND DIFFUSION

Streamline diffusion introduces artificial diffusion in the streamline direction. This is often enough to obtain a smooth numerical solution if the exact solution of Equation 3-2 (or Equation 3-6) does not contain any discontinuities. At sharp gradients, however, undershoots and overshoots can occur in the numerical solutions (see Figure 3-21). Crosswind diffusion addresses these spurious oscillations by adding diffusion orthogonal to the streamline direction—that is, in the crosswind direction.

Crosswind diffusion methods are consistent, but they are also nonlinear. This means that the discrete equation system becomes nonlinear even if the original equation (Equation 3-2 or Equation 3-6) is linear, which can increase the computational cost.

Use crosswind diffusion if it is important to avoid undershoots or overshoots. Typical examples are concentrations that must not become negative and mass fractions that must be between zero and one.

[1] The streamline diffusion stabilization in COMSOL is GLS but without any viscous terms in the test operator in the stabilization term.

The crosswind diffusion option adds a weak contribution as suggested in Ref. 5. For the scalar example here, the term reads

$$-\nu^h \frac{\partial c}{\partial x_i} g^{ij} \frac{\partial c}{\partial x_j}$$

where g^{ij} is the covariant metric tensor. The coefficient ν^h is for Navier-Stokes systems a modified version of the Hughes-Mallet (HM) formulation of Ref. 6. In the scalar case, the modified HM formulation reduces effectively to the form suggested in Ref. 6. Additionally, Ref. 7 suggests to reduce ν^h for higher-order elements. The COMSOL formulation multiplies ν^h with a factor

$$(\sqrt{2})^{1-N}$$

where N is the shape function order.

Figure 3-22 shows the example problem (Equation 3-4) solved using streamline diffusion and crosswind diffusion. Oscillations at the boundary layers are almost completely removed (compare with Figure 3-21), but it has been achieved by the introduction of some extra diffusion. In general, crosswind diffusion tries to smear out the boundary layer so that it becomes just wide enough to be resolved on the mesh (Figure 3-17). To obtain a sharper solution and remove the last oscillations, the mesh needs to be refined locally at the boundary layers.

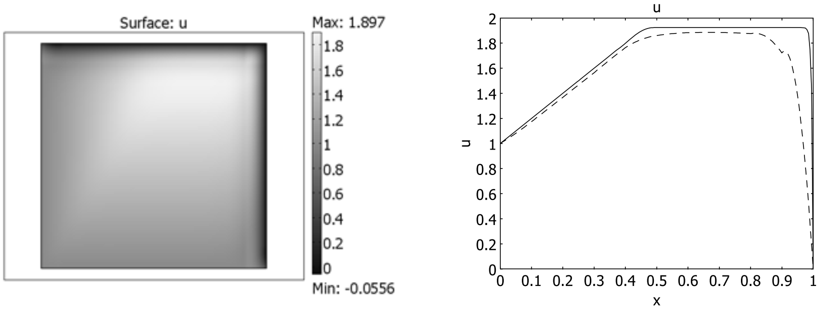


Figure 3-22: Equation 3-4 solved using streamline diffusion and crosswind diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).

References for Stabilization Techniques

1. O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu, *The Finite Element Method for Fluid Dynamics*, 6th ed., Elsevier, 2005.

2. R. Codina, “Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation,” *Comput. Methods Appl. Mech. Engrg.*, vol. 156, pp. 185–210, 1998.

3. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Student literature, 1987.

4. G. Hauke, “A Simple Subgrid Scale Stabilized Method for the Advection-Diffusion-Reaction Equation,” *Comput. Methods Appl. Mech. Engrg.*, vol. 191, pp. 2925–2947, 2002.

5. G. Hauke and T.J.R. Hughes, “A comparative study of different sets of variables for solving compressible and incompressible flows,” *Computer Methods in Applied Mechanics and Engineering*, vol. 153, pp. 1–44, 1998.

6. E.G.D. do Carmo and A.C. Galeão, “Feedback Petrov-Galerkin methods for convection-dominated problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 88, pp. 1–16, 1991.

7. E.G.D. do Carmo and G.B. Alvarez, “A new upwind function in stabilized finite element formulations, using linear and quadratic elements for scalar convection-diffusion problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 2383–2402, 2004.

Using Units

COMSOL supports a number of consistent unit systems, including the SI unit system, which is the default unit system. The physics interface displays the unit for the physical quantities entered in the selected unit system, but by Using Standard Unit Prefixes and Syntax you can use any available and applicable unit or SI prefix to define your input quantities. In addition to SI units, many English units and units from the CGS (or cgs) system are also available, regardless of the unit system used in the model. All data in the material databases and Material Library product use SI units with declared units using the unit syntax (see Materials). This makes it possible to use the material data also in models with non-SI unit systems. Regardless of the selected unit system, you can always choose from a list of applicable units for plotting and results evaluation.

In the unit tables, “N/A” means that no unit symbol is available.

- Unit Systems in COMSOL
- Indication of Unexpected, Unknown, or Inconsistent Units
- Setting the Unit System for Models

Using Standard Unit Prefixes and Syntax

STANDARD UNIT PREFIXES

For SI units you can scale data using the standard prefixes for powers of 10—*kilo*, *mega*, *milli*, or *micro*, for example. Either the full prefix or the symbol can be used, but you must use the same form for the prefix and the unit—that is, `[milliampere]` and `[mA]` are valid but not `[mampere]` or `[milliA]`). In the settings windows for plotting and numerical results, the **Unit** list contains the SI unit for the quantity, including the most common prefixes. The lists also contain applicable non-SI units, which in a few cases also support these prefixes.

Use Table 3-6 as a guide for the format to enter.

TABLE 3-6: SI PREFIXES

FULL PREFIX	SYMBOL	FACTOR
yotta	Y	10^{24}
zetta	Z	10^{21}
exa	E	10^{18}
peta	P	10^{15}
tera	T	10^{12}
giga	G	10^{9}
mega	M	10^{6}
kilo	k	10^{3}
hekto	h	10^{2}
deca	da	10^{1}
deci	d	10^{-1}
centi	c	10^{-2}
milli	m	10^{-3}

TABLE 3-6: SI PREFIXES

FULL PREFIX	SYMBOL	FACTOR
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}
atto	a	10^{-18}
zepto	z	10^{-21}
yocto	y	10^{-24}

STANDARD UNIT SYNTAX

You can use the unit syntax to specify a quantity with any applicable unit. To do so, append the unit to any constant or variable in a model using a syntax where you enclose the unit in brackets, for example, `200[ft]` and `3e6[kg/m^3]`.

Both the name and the symbol can be used for a unit. For example, `2.4[ampere]` and `2.4[A]` are both valid to indicate an electric current in SI units. The SI units can also contain standard prefixes. Appending a unit means that you *multiply* the constant or variable to the left of the unit declaration with this unit. This multiplication takes precedence over other operators so, for example, `1/2[m]` evaluates to 0.5 m^{-1} (`0.5[1/m]`) whereas both `(1/2)[m]` and `1/2*1[m]` evaluate to 50 cm (`0.5[m]` or `50[cm]`). Also, if `L` is a variable defined as `2[m]`, `L[1/s]` evaluates to `2[m/s]`.

The following examples show how to apply the unit syntax:

- Adding two quantities of the same kind that use different units: `0.5[ft]+33[mm]`. COMSOL converts the result to the base unit system's length unit.
- Using multiplication with a unit to get consistent units for two quantities that you want to add, for example, `14[kg]+ht.rho[m^3]`, which works if `ht.rho` represents the density for a heat transfer model. You can also concatenate several units, for example, `3.6[N][m]`, which is equivalent to typing `3.6[N*m]` and evaluates to `3.6 N·m`.

For unit names with spaces and hyphens, such as *British thermal unit* and *pound-force*, only use the symbols when declaring units.

It is possible to add constants (without units) to any quantity. COMSOL then assumes that this value has the same unit as that quantity (as indicated in the settings window).

All data in the material databases and Material Library product use SI units and this unit syntax.

DECLARING UNITS FOR PARAMETERS, VARIABLES, AND FUNCTIONS

It is important to be aware of the following aspects of unit handling.

When using parameters, variables, and functions in expressions:

- If user-defined parameters or variables are used in the physics, it is good practice to use the unit syntax to define them. The settings windows for parameters and variables display the resulting unit, in the models base unit system, of user-defined parameters and variables. It is important to verify that the variables have the expected unit before using them in the physics settings. The unit of parameters and variables is otherwise undefined.

- Most user-defined and built-in functions expect dimensionless inputs and outputs, so it is good practice to use make inputs, such as time, dimensionless using unit syntax. If the input is not dimensionless, COMSOL marks the expression in an orange color and reports an unexpected unit of input. For example, to use the time *t* as input to a Rectangle function `rect1`, use `[1/s]` to make the input dimensionless: `rect1(t[1/s])`.
- Using properties with undefined units in a model does not affect the numerical results during the analysis, but undefined units are required in the results and visualization stages—expressions involving such parameters and variables are also unitless.
- If other units than the base unit system's units are used or if SI prefixes are included, the conversion to base units also affects the value (quantity) using a scaling factor (and an offset for temperature units). The **Value** column in a **Parameter** settings window displays the quantity and unit in the base unit system so that you can see the result of the unit conversion. For example, a parameter is defined as `3[ft]`, the result in the **Value** column is **0.9144 m** if the base unit system is SI.

SI Base, Derived, and Other Units

The SI units form an internationally accepted system with seven units for base quantities and a large number of derived units. Use the symbols for these and other units when declaring units in COMSOL (for example, `10[m/s]` uses the SI unit for velocity).

- Table 3-7 lists the SI units for the seven base quantities.
- Table 3-8 lists the SI derived units supported in COMSOL.
- Table 3-9 lists additional units available in COMSOL regardless of the unit system in the model. If more than one name or symbol is available, use any of them, except when names contain more than one word or a hyphen. See also the tables with special units for other unit systems than the SI system; special units that are not listed in Table 3-9 are only available when using such non-SI unit systems.
- Table 3-10 lists other SI derived units without special names or symbols.

TABLE 3-7: BASE SI UNITS

BASE QUANTITY	UNIT NAME	SYMBOL
length	meter, metre*	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
temperature	kelvin**	K
amount of substance	mole	mol
luminous intensity	candela	cd

* See About Editing Geometry Length and Angular Units

**See About Temperature Units

TABLE 3-8: SI DERIVED UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
absorbed dose	gray	Gy
capacitance	farad	F
conductance	siemens	S
dose equivalent	sievert	Sv
electric charge	coulomb	C

TABLE 3-8: SI DERIVED UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
electric resistance, impedance, reactance	ohm*	Ω
electric potential difference, voltage	volt	V
energy, work, heat	joule	J
force, weight	newton	N
frequency	hertz	Hz
inductance	henry	H
magnetic flux	weber	Wb
magnetic flux density, magnetic induction	tesla	T
plane angle	radian	rad
power	watt	W
pressure	pascal	Pa
* See the additional notes following the next table.		

TABLE 3-9: ADDITIONAL UNITS IN COMSOL

QUANTITY	NAME	SYMBOLS	VALUE
acceleration	galileo	Gal	0.01 m/s^2
dipole moment	debye	D	3.33564095·10^{-30} C·m
dynamic viscosity	poise	P	0.1 Pa·s
energy	British thermal unit*	BTU, Btu	1055.05585 J
energy	calorie*	cal	4.184 J
energy	electronvolt	eV	1.6021765314·10^{-19} J
energy	erg	erg	10^{-7} J
force	dyne	dyn	10^{-5} N
force	kilopond*	kp, kpf	9.80665 N
force	poundal	pdl	0.138254954376 N
force	pound-force	lbf	4.4482216152605 N
frequency	rpm	RPM	1/60 Hz
length	angstrom	Å	10^{-10} m
length	inch	in	0.0254 m
length	foot	ft	0.3048 m
length	mile*	mi	1609.344 m
length	microinch	uin	0.0254·10^{-6} m
length	milliinch	mil, thou	0.0254·10^{-3} m
length	nautical mile*, nautimile	nmi	1852 m
length	yard	yd	0.9144 m
magnetic field strength	oersted	Oe	10^3/(4·π) A/m
magnetic flux density	gauss	G	10^{-4} T
mass	atomic mass unit, dalton	u, amu, Da	1.660538782·10^{-27} kg
mass	gram	g	0,001 kg
mass	pound, pound-mass	lb, lbm	0.45359237 kg
mass	stone	st	6.35029318 kg

TABLE 3-9: ADDITIONAL UNITS IN COMSOL

QUANTITY	NAME	SYMBOLS	VALUE
mass	slug	slug	approx. 14.5939 kg
mass	ton, tonne	t	1000 kg
permeability	millidarcy*	mD	$9.869233\cdot10^{-16}$ m^2
plane angle	degree	deg	$\pi/180$
pressure	atmosphere	atm	101325 Pa
pressure	bar	bar	100000 Pa
pressure	barye	ba	0.1 Pa
pressure	kilopound per square inch*	ksi	$6.894757\cdot10^{6}$ Pa (1000 psi)
pressure	psi	psi	$6.894757\cdot10^{3}$ Pa
pressure	torr	Torr, mmHg	133.322 Pa
pressure	inches water*	inAq, inH2O	249.089 Pa
speed	mph, MPH	mph	0.44704 m/s
speed	knot*	knot	1852 km/h (approx. 0.614 m/s)
temperature	Celsius**	degC	T+273.15
temperature	Fahrenheit**	degF	5/9·T+459.67
temperature	Rankine**	R, Ra	5/9·T
time	year*	a, yr	31556952 s
time	day	d	86400 s
time	hour	h	3600 s
time	minute	min	60 s
volume	gallon*	gal	0.003785411784 m^3
volume	imperialgallon	impgal	0.00454609 m^3
volume	liter, litre	L, l	0.001 m^3
volume	pint*	pt	0.000473176473 m^3
volume	quart*	qt	0.000946352946 m^3
volumetric flow rate	cubic feet per minute	CFM, cfm	$4.719474\cdot10^{-4}$ m^3/s

* See the additional notes following this table.

** See About Temperature Units

ADDITIONAL NOTES ABOUT UNITS IN Table 3-8 **AND** Table 3-9

UNIT NAME	NOTE
British thermal unit	An energy unit defined as the amount of heat required to raise the temperature of one pound (pound-mass) of water by one degree from 60° to 61° Fahrenheit at a constant pressure of one atmosphere. Refer to the British thermal unit using the symbol only (Btu or BTU): for example, `0.28[Btu/(h*in*degF)]` for a thermal conductivity.
calorie	Small calorie or gram calorie, which equals 4.184 J. A large calorie or kilogram calorie is 1000 calories (4.184 kJ). Use [kcal] for large calories.
kilopond	The kilopond (kp) or kilogram-force (kpf) is a gravitational metric unit of force. Refer to this unit using kilopond, kp, or kpf only.

ADDITIONAL NOTES ABOUT UNITS IN Table 3-8 AND Table 3-9

UNIT NAME	NOTE
kilopound per square inch	The kilopound per square inch (ksi) is a scaled pressure unit derived from psi (1 ksi is equal to 1000 psi). Refer to this unit using the symbol only (ksi).
millidarcy (mD)	Widely used for permeability in petroleum engineering. Typical values for the permeability of porous media are in the range of a few to a few hundred mD. The symbol D represents the debye, a unit for the magnetic dipole moment, and not the darcy unit.
mile	The international statute mile, which equals 1609.344 m.
nautimile	The nautical mile equals 1852 m.
ohm	To declare the SI unit for electric resistance, ohm, use [ohm]. COMSOL then displays this as Ω
inches water	The value of 1 inch of water is defined as the pressure exerted by one inch of water for a pure water density of 1000 kg/m^3 at 4 degrees Celsius and standard gravity of 9.80665 m/s^2. Refer to this unit using the symbol only (inH2O or inAq).
knot	The same as nautical miles per hour.
year	A Gregorian year, which equals 365.2425 days.
gallon (gal)	This is the U.S. liquid gallon which equals 0.003785411784 m^3; the Imperial (UK) gallon (imperialgallon, impgal) is equal to 0.00454609 m^3.
pint and quart	The U.S. liquid pint and U.S. liquid quart, respectively.

TABLE 3-10: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

DERIVED QUANTITY	NAME	SYMBOL
acceleration	meter per second squared	m/s^2
amount-of-substance concentration	mole per cubic meter	mol/m^3
area	square meter	m^2
current density	ampere per square meter	A/m^2
heat capacity, specific heat	joule per kilogram kelvin	J/(kg·K)
magnetic field strength	ampere per meter	A/m
mass density	kilogram per cubic meter	kg/m^3
permeability	henry per meter	H/m
speed, velocity	meter per second	m/s
wave number	reciprocal meter	m^{-1}
volume	cubic meter	m^3

Special British Engineering Units

The base units in the British engineering unit system are identical to the SI units with the following exceptions:

TABLE 3-11: SPECIAL BASE UNITS IN THE BRITISH ENGINEERING UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	slug	N/A
temperature	Fahrenheit	degF

There is one derived unit that differs from the corresponding SI unit:

TABLE 3-12: DERIVED BRITISH ENGINEERING UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
force	pound-force	lbf

The British thermal unit is also available as Btu or BTU.

If the British engineering unit system is the base unit system, COMSOL constructs derived units from the applicable SI base units and the units listed in Table 3-11 and Table 3-12. This means, for example, that the unit for voltage displayed in the physics interface is lbf·ft/As rather than V (volt). In a text field that expects a voltage as input, you need to use the unit syntax when entering a numerical value, for example, `10[V]`.

Special CGSA Units

The base units in the CGSA unit system are identical to the SI units with the following exceptions:

TABLE 3-13: SPECIAL BASE UNITS IN THE CGSA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g

The CGSA unit system includes the following derived units that differ from the corresponding SI units:

TABLE 3-14: DERIVED CGSA UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
energy	erg	N/A
force	dyne	dyn
pressure	barye	N/A
speed	kyne	N/A

Special EMU Units

The base units in the EMU unit system are identical to the SI units with the following exceptions:

TABLE 3-15: SPECIAL BASE UNITS IN THE EMU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
electric current	abampere, biot	N/A

The EMU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 3-16: DERIVED EMU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
capacitance	abfarad	N/A
conductance	absiemens	N/A

TABLE 3-16: DERIVED EMU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
electric charge	abcoulomb	N/A
electric resistance	abohm	N/A
electric potential difference, voltage	abvolt	N/A
energy	erg	N/A
force	dyne	dyn
inductance	abhenry	N/A
magnetic flux	abweber	N/A
magnetic flux density	abtesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special ESU Units

The base units in the ESU unit system are identical to the SI units with the following exceptions:

TABLE 3-17: SPECIAL BASE UNITS IN THE ESU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
electric current	statampere, franklin	N/A

The ESU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 3-18: DERIVED ESU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
capacitance	statfarad	N/A
conductance	statsiemens	N/A
electric charge	statcoulomb	N/A
electric resistance	statohm	N/A
electric potential difference, voltage	statvolt	N/A
energy	erg	N/A
force	dyne	dyn
inductance	stathenry	N/A
magnetic flux	statweber	N/A
magnetic flux density	stattesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special FPS Units

The base units in the FPS unit system are identical to the SI units with the following exceptions:

TABLE 3-19: SPECIAL BASE UNITS IN THE FPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	pound	lb
temperature	Fahrenheit	degF

There is one derived unit that differs from the corresponding SI unit:

TABLE 3-20: DERIVED FPS UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
force	poundal	N/A

Special IPS Units

The base units in the IPS unit system are identical to the SI units with the following exceptions:

TABLE 3-21: SPECIAL BASE UNITS IN THE IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	pound	lb
temperature	Fahrenheit	degF

Special MPa Units

The base units in the MPa unit system are identical to the SI units with the following exceptions:

TABLE 3-22: SPECIAL BASE UNITS IN THE MPA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	millimeter	mm
mass	tonne, ton	t

There is one derived unit that differs from the corresponding SI unit:

TABLE 3-23: DERIVED MPA UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
pressure	megapascal	MPa

Special Gravitational IPS Units

The base units in the Gravitational IPS unit system are identical to the SI units with the following exceptions:

TABLE 3-24: SPECIAL BASE UNITS IN THE GRAVITATIONAL IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	GIPS_mass	N/A
temperature	Fahrenheit	degF

The following derived units differ from the corresponding SI units:

TABLE 3-25: DERIVED GRAVITATIONAL IPS UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
force	pound-force	lbf
pressure	psi	psi

Switching Unit System

If the unit system is switched during modeling, COMSOL does not convert the data in the model (except for length units and angular units in the geometry if specified). All physical constants and data in the material libraries are in SI units and defined using the unit syntax, so you can use them with any unit system because COMSOL converts these values to the corresponding values in the model's unit system.

> The units of other input data must be declared for the model using the unit syntax (or manually convert the numerical values to the new unit system).

About Temperature Units

The relationship between different temperature units involve an offset in addition to the usual scale factor. The offset is often not important to the physics equations because these equations are concerned only with temperature differences. There are, however, some cases where an absolute or thermodynamic temperature measure must be used. One example is the Stefan-Boltzmann law for blackbody radiation used in radiation boundary conditions.

The SI unit system uses the kelvin, which is an absolute temperature, as the basic unit of temperature. English unit systems use degree Fahrenheit as the basic unit of temperature, which, because the Fahrenheit scale is not absolute, is fine for most purposes except radiation. For such purposes, the Rankine scale provides the corresponding absolute temperature unit. See Table 3-26 for a list of acceptable unit syntax.

TABLE 3-26: TEMPERATURE UNITS

SCALE	UNIT
Celsius	`[degC]`
Fahrenheit	`[degF]`
Kelvin	`[K]`
Rankine	`[R] or [Ra]`

DIFFERENTIAL VS. ABSOLUTE TEMPERATURE

If the dimension of an expression that includes a unit is temperature or 1/temperature, COMSOL interprets the dimension as an absolute temperature. If the dimension is something other than temperature but the unit expression includes temperature, the temperature is a differential temperature; that is, COMSOL uses no offset when converting between different temperature units.

The following examples show how the unit conversion works for different expressions that include temperature units:

- `100[degC]` is an expression that has temperature as the dimension. COMSOL interprets it as an absolute temperature and evaluates it as 373.15 K.
- `373.15[1/K]` is interpreted as an absolute inverse temperature (but no conversion is necessary from kelvin to kelvin).
- `373.15[1/degC]` evaluates to `100[1/K]` using the offset of 273.15 degrees between kelvin and degrees Celsius.

- `100[degC/K]` is dimensionless, and the temperature is therefore a differential temperature; that is, the result is 100 because the conversion uses no offset.
- To make COMSOL interpret `100[degC/K]` as an absolute temperature, split the expression using two separate expressions such as `100[degC]*1[1/K]`, which equals 373.15. This is also what occurs when you use a variable (`TC`, for example) defined as `100[degC]`. `TC[1/K]` is then also two expressions where both are interpreted as absolute temperature.

About Editing Geometry Length and Angular Units

The default units are meters for length and degrees for angles. For many applications an independent length unit for the geometry might be required. For example, if the model describes a MEMS device, the natural length unit might be µm (micrometers), or the geometry imported from a CAD file might use another unit than meters. It can also be useful to specify the angular unit in radians instead.

The length unit for the geometry does not affect the units that include length in the physics interfaces or any other part of COMSOL.

1 Create or open a model file.

2 In the **Model Builder**, under a **Component** node, click the **Geometry** node.

3 Under **Units**, select a **Length unit** from the list.

4 Select an **Angular unit**—**Degrees** or **Radians**.

5 Select the **Scale values when changing units** check box to automatically scale for dimensions in the existing geometry.

6 Enter a **Default relative repair tolerance** and select a **Geometry representation.**

When importing 3D CAD geometries, you can choose to use the length unit from the CAD file or the length unit from COMSOL.

The Geometry Node

Units and Space Dimensions

Most physics interfaces support 2D (and in some cases also 1D) models in addition to 3D models. The units for intensive physical quantities such as density in the physics interfaces are the same regardless of the space dimension (for density, kg/m^3 in SI units). This makes it possible to use common material property values also in models with other space dimensions than 3D using their well-known, physical units regardless of the dimension you are modeling in. In planar 2D, this means that the implementation includes an implicit unit depth in the out-of-plane direction, except for some physics interfaces (for solid mechanics and electric currents, for example), where the thickness is a user-defined property that defines the volume of the model domain. In axisymmetric models, the volume of the domain is defined by the 2D cross section in the *rz*-plane that is the geometry you define for such models. The volume that it defines is the area of the 2D cross section integrated a full 360 degrees in the circumferential direction.

4

Customizing the COMSOL Desktop

This chapter describes a variety of tasks that can be done to organize and simplify the model building process. For example, set the layout and other features of the COMSOL Desktop using the Preferences settings, change the language and fonts, restrict or allow features to display based on license, or learn about how to edit node names.

Customizing a Model

A variety of tasks can be done to organize and simplify the model building process.

- Customizing the Desktop Layout
- Changing Fonts and the Desktop Language
- Editing Node Properties, Names, and Identifiers
- Grouping Nodes by Space Dimension and Type
- Setting the Unit System for Models
- Checking and Controlling Products and Licenses Used

- Preferences Settings
- The Model Libraries Window
- Advanced Physics Sections
- The COMSOL Desktop

Customizing the Desktop Layout

To customize the COMSOL Desktop environment, you can rearrange the windows by moving, resizing, detaching, or docking each window (see Adjusting Window Location and Size on the Desktop). Predefined layouts are also available and selected from the **Desktop Layout** menu where you can adjust to a widescreen or regular layout or reset it to the default.

You can also adjust the fonts and the language. See Changing Fonts and the Desktop Language.

CHANGING AND RESETTING THE DESKTOP LAYOUT

The COMSOL Desktop layout can be set to widescreen or regular screen, or you can reset it to its default settings. Resetting can be useful after you have been moving or resizing the windows and you want to quickly return to the default. The default settings are restored either for a widescreen layout or a regular screen layout depending on the monitor. Also see The COMSOL Desktop Menus and Toolbars.

From the **Desktop Layout** menu, select one of the following:

- **Widescreen Layout**: suitable for widescreen monitors. The **Model Builder** window and the settings window display side by side.
- **Regular Screen Layout**: suitable for monitors with a regular screen (4:3). The **Model Builder** window displays on top of the settings window.

Win

To access the menu, on the **Home** ribbon, click **Layout>Desktop Layout** to choose a layout, or choose to **Reset Desktop**.

Mac

Linux

To access the menu, select a layout from **Windows>Desktop Layout**. To reset the desktop, select **Reset Desktop** or click the **Reset desktop** ( ) button on the main toolbar.

Changing Fonts and the Desktop Language

COMSOL uses a default font for texts in plots such as axis labels and titles. This might be necessary to display non-Latin characters such as Chinese and Japanese characters. You can also change the Desktop language. You can make these changes using The Preferences Dialog Box.

CHANGING THE FONT FOR PLOT LABELS AND TITLES

The **Font** option is for the text that displays for plots in the **Graphics** window. The change is applied when creating a new model. You can also change the setting used for a model from the root node's settings window. In the **Preferences** dialog box, click **Graphics**.

1 Under **Font**, select a font **Family** from the list and enter a font **Size** (in points). The default is to use a predefined **default** font with a font **Size** of 9 points. Depending on the operating system and the installed fonts on the computer, you can select from a number of other font families.

2 Click **OK**. The program stores the specified font family and size as a preference setting, so you only have to change it once.

- Plot Titles for Plot Groups and Plot Types
- The Root Window

CHANGING THE COMSOL DESKTOP LANGUAGE

1 In the **Preferences** dialog box, click **General**.

2 Select an available **Language** for the graphical user interface (GUI). The following languages are available: Traditional Chinese, English, French, German, Italian, Japanese, Korean, and Spanish.

3 Click **OK**. A message displays indicating that COMSOL must be restarted for the changes to take effect. Click **OK** again, exit and re-open COMSOL to display the GUI in the selected language.

If you selected a language during the COMSOL installation, that language becomes the default language when you first start COMSOL Multiphysics.

Editing Node Properties, Names, and Identifiers

All nodes, except container nodes, have a common set of node properties, some of which can be changed and some system generated properties that cannot be edited. The **Root** node has additional information that provides an overview of the complete model file. The settings window for the Root node also includes a **Node Properties** section with additional information about the model file. The **Model Thumbnail** is set from The Root Window.

- The Root Window
- To learn about how some of these properties can be viewed, see Viewing Node Names, Identifiers, Types, and Tags.
- Variable Naming Convention and Namespace

EDITING A COMPONENT IDENTIFIER FOR USE WITH VARIABLES

Use a *component identifier* to access variables throughout the model. The identifier is part of the full reference to variables (for example, when referring to variables in another model). To edit a component identifier, in the **Model Builder**, click a **Component** node. The **Component** node's settings window opens. Edit the default identifiers (**comp1**, **comp2**, and so on) as required in the **Component Identifier** field. See The Node Settings Windows (Figure 3-3) for an example.

RENAMING A NODE

To rename a node in the **Model Builder** (except container nodes with fixed names such as **Global Definitions**), click a node and select **Rename** ( ) or press F2. Enter a **New name** and click **OK**. The **Name** is both updated in the **Model Builder** and in the **Node Properties** section.

THE NODE PROPERTIES SETTINGS WINDOW

In the **Model Builder**, right-click a node (except "container nodes" such as **Materials**) and select **Properties** from the context menu (see Figure 3-4). The **Properties** window for that specific node replaces the settings window. The node properties vary by node type.

Node Properties

This section contains these fields:

- The **Name** field defaults to a system name for the node. Edit as required as a description for the node.
- The **Tag** is a unique system-defined tag for the node, which cannot be changed and is mainly used when using the COMSOL API and the optional LiveLink™ *for* MATLAB®.
- **Identifier**: For physics interfaces, this is the node identifier.
- The **Created** field is system generated and shows the node creation date and time.
- The **Author** field contains the name of the author (creator) of the node. In addition to editing the author name manually, you can make a change for all nodes that you add later on from The Preferences Dialog Box.
- The **Version** and **Description** fields are empty by default. Enter version numbers or comments to track model changes or changes to specific node contents.

Returning to the Settings Window

When you are finished editing the properties, right-click the node again and select **Settings** to return to the settings window (or click another node and then click the node again).

The default setting, found under **Model Builder** in the The Preferences Dialog Box section, disables grouping in new models.

The **Group by Space Dimension** (for physics interface nodes), **Group by Type** (for **Definitions** nodes), or **Ungroup** options are available from the context menu for these features:

- **Global Definitions**—nodes can be grouped by type: **Variables**, **Functions**, or **Groups**.
- **Definitions**—nodes can be grouped by type: **Functions**, **Variables**, **Selections**, **Probes**, **Component Couplings**, **Coordinate Systems**, **Pairs**, and **Domain Properties** (this includes Perfectly Matched Layers and Infinite Elements).
- For any physics interface, nodes can be grouped by space dimension—that is, by geometric entity level: **Domains**, **Boundaries**, **Edges** (3D only), or **Points**.

For physics interfaces, when either of these options is selected, the way the nodes are organized changes in the Model Builder and when you right-click to view the context menu.

GROUPING NODES BY SPACE DIMENSION

The default is **Ungroup** in the **Model Builder** for all new models. Right-click and select **Group by Space Dimension** () from the context menu to both group the nodes in the Model Builder and in the context menu (see Figure 4-1).

In Figure 4-1 for the **Pressure Acoustics, Frequency Domain** physics and when **Group by Space Dimension** is selected, the default **Pressure Acoustic Model** and **Initial Values** subnodes are included under the **Domains** node, and four boundary level nodes are included under **Boundaries**. However, there are no nodes under **Edges** or **Points** even though these nodes display in the **Model Builder**.

When **Ungroup** () is selected from the context menu (or the default is kept), the tree is flattened and all nodes are ungrouped. Only default physics nodes and user-added nodes are included in the **Model Builder** sequence.

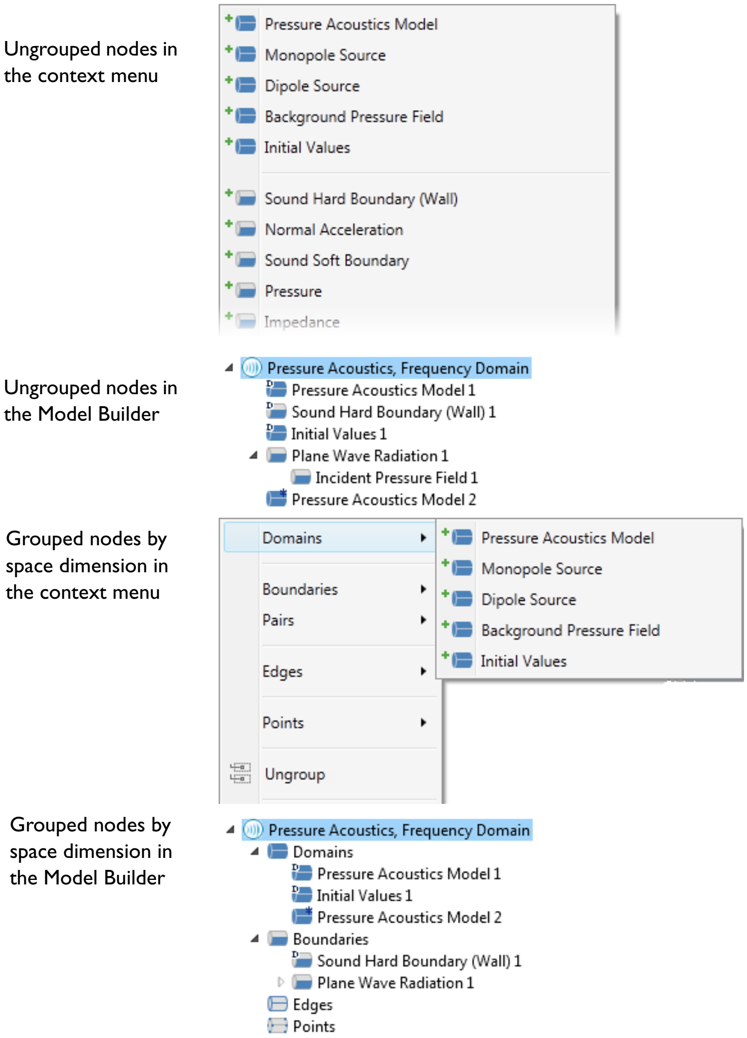

Ungrouped nodes in the context menu
Pressure Acoustics Model
Monopole Source
Dipole Source
Background Pressure Field
Initial Values
Sound Hard Boundary (Wall)
Normal Acceleration
Sound Soft Boundary
Pressure
Impedance
Ungrouped nodes in the Model Builder
Pressure Acoustics, Frequency Domain
Pressure Acoustics Model 1
Sound Hard Boundary (Wall) 1
Initial Values 1
Plane Wave Radiation 1
Incident Pressure Field 1
Pressure Acoustics Model 2
Grouped nodes by space dimension in the context menu
Domains
Boundaries
Pairs
Edges
Points
Ungroup
Pressure Acoustics Model
Monopole Source
Dipole Source
Background Pressure Field
Initial Values
Grouped nodes by space dimension in the Model Builder
Pressure Acoustics, Frequency Domain
Domains
Pressure Acoustics Model 1
Initial Values 1
Pressure Acoustics Model 2
Boundaries
Sound Hard Boundary (Wall) 1
Plane Wave Radiation 1
Edges
Points

Figure 4-1: The context menu and Model Builder sequence for physics nodes when Group by Space Dimension or Ungroup is selected. For the Global Definitions and Definitions nodes, the same principles apply but the nodes are grouped by type instead.

GROUPING NODES BY TYPE

The same principles apply for the **Global Definitions** and **Definitions** nodes as for the physics nodes, except the nodes are grouped by type, that is:

- Under **Global Definitions** the types are **Variables**, **Functions**, or **Groups**.
- Under **Definitions** the types are **Functions**, **Variables**, **Selections**, **Probes**, **Component Couplings**, **Coordinate Systems**, **Pairs**, and **Domain Properties**.

The default is **Ungroup** in the **Model Builder** for all new models. Right-click and select **Group by Type** ( ) from the context menu to both group the nodes in the Model Builder and in the context menu.

When **Ungroup** ( ) is selected from the context menu (or the default is kept), the tree is flattened and all nodes are ungrouped. Only default **Definitions** or **Global Definitions** nodes and user-added nodes are included in the **Model**

Builder sequence as in Figure 4-2.

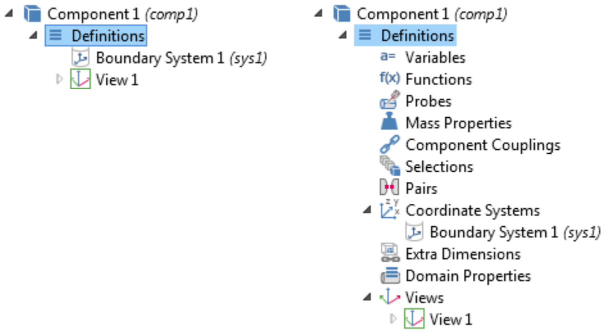


Figure 4-2: When Ungroup (left) or Group by Type (right) is selected from the context menu for the Definitions node. When grouped by type, all categories display underneath the node even if it does not contain a feature node.

- Opening Context Menus and Adding Nodes
- The Physics Nodes
- The Physics Interfaces
- Building a COMSOL Model

Setting the Unit System for Models

SETTING THE UNIT SYSTEM ON THE GLOBAL LEVEL

To set the unit system for the entire MPH-file on a global level:

1 In the **Model Builder**, click the root node (the top node in the model tree). The root node's name is the name of the MPH-file or **Untitled.mph** before you have saved your work.

2 In the root node's settings window, select the unit system from the list in the **Unit System** section or **None** to turn off unit support.

SETTING THE UNIT SYSTEM FOR INDIVIDUAL MODELS

By default, all components in a model use the same global unit system, but it is possible to use different unit systems in each component. To do so, follow these steps:

1 In the **Model Builder**, click the top node for a **Component** branch (**Component 1**, for example).

2 In the **Component** node's settings window, locate the **Component Settings** section.

3 Select the **Override global system** check box, and then select the unit system from the list of unit systems that becomes available.

To disable unit support in a model, choose **None** from the list in the **Unit System** section in the root node's settings window and make sure that the corresponding setting for each component is **Same as global system**. If unit support is turned off only for some components (or if the settings at global and component level differ for a model with a single component) unexpected side effects can occur.

- Using Units
- The Root Window

Checking and Controlling Products and Licenses Used

Open the **Licensed and Used Products** dialog box to view a list of licenses or to block the use of a product. Blocking a license can be useful for consultants who want to duplicate a client's environment while building a model or when collaborating with other users who do not have access to the same set of COMSOL products. You can also use these settings to prevent the use of a module when sharing a floating-network license, for example.

By default the use of all products is active and the check boxes for all products are selected. The licenses currently in use are unavailable (you cannot block the use of products with functionality already in use without restarting COMSOL).

To open the **Licensed and Used Products** dialog box:

- From the **File** menu, select **Licensed and Used Products** (🔑) (Windows users). You can also customize the Quick Access Toolbar and then click the button. See Windows Toolbars and Menus.
- From the main menu select **Options>Licensed and Used Products** (🔑) (Mac and Linux users).

Click to clear the check box next to a product to hide or block it from use. Click **Select All** to activate all products. Click **Deselect All** to block all products (except the ones that are already in use). Click **OK** to save the changes or **Cancel** to discard any changes and close the window.

PRODUCT INFORMATION

Click the **Product Information** button to go to the product information pages on the COMSOL website, where you find information about all COMSOL products.

You can also get information about the licensed products from The About COMSOL Multiphysics Box.

BORROW A LICENSE

If you have a floating network license (FNL) or a class kit license (CKL) and your license file has been enabled for borrowing, click **Borrow** to open the **Borrow Licenses** dialog box and borrow licenses from the license server. Select the licenses you want to borrow from the list and specify the number of days you want to keep them. Click **OK** to save.

Remember that other users cannot use the licenses that you have checked out. If you try to borrow a license that has already been borrowed, you receive an error message that shows for which products the license has been borrowed.

Preferences Settings

The Preferences Dialog Box

To make changes to how items are displayed throughout COMSOL edit the following settings in the **Preferences** dialog box as required.

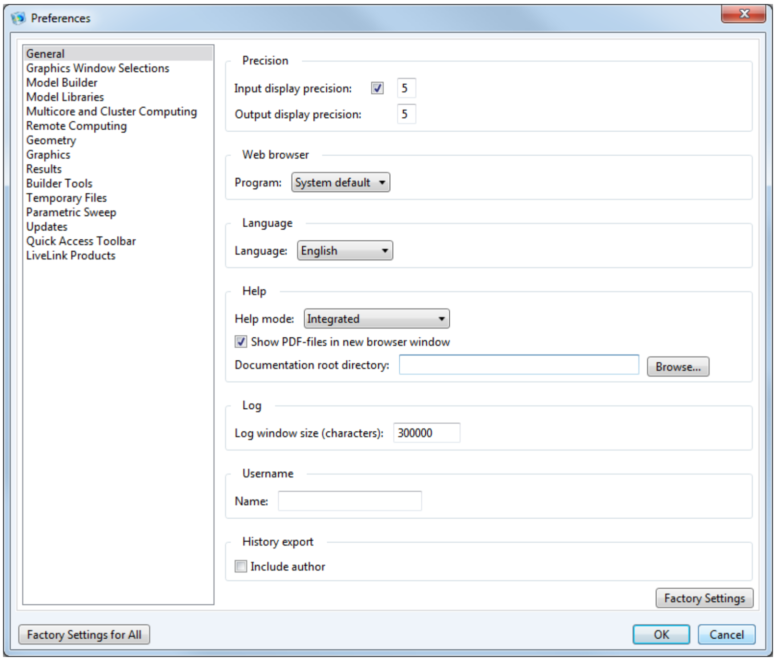
To open the **Preferences** dialog box:

- Windows users: From the **File** menu, select **Preferences** ( ) You can also customize the Quick Access Toolbar and then click the button. See Windows Toolbars and Menus.
- Cross-platform (Mac and Linux) users: From the main menu select **Options>Preferences** ( ).

> !
>
> For some setting changes, a message window displays to tell you that COMSOL needs to be restarted for the changes to take effect. For each preference settings window, you can click the **Factory Settings** button to restore the factory default values, or click **Factory Settings for All** to reset all the preferences to the factory default.

As shown in Figure 4-3, the following are available in the **Preferences** dialog box:

- General
- Graphics Window Selections
- Model Builder
- Model Libraries
- Multicore and Cluster Computing
- Remote Computing
- Geometry
- Graphics and Plot Windows
- Results
- Builder Tools
- Temporary Files
- Parametric Sweep
- Updates
- Quick Access Toolbar (Windows only)
- LiveLink Products



Precision

Input display precision: ☑ 5

Output display precision: 5

Web browser

Program: System default

Language

Language: English

Help

Help mode: Integrated

☑ Show PDF-files in new browser window

Documentation root directory:

Log

Log window size (characters): 300000

Username

Name:

History export

☐ Include author

Figure 4-3: The Preferences dialog box.

Advanced Physics Sections

To display the functionality described in this section, click the **Show** button (👁) on the **Model Builder** and then select the applicable option. To always display sections on a physics interface settings window, click the **Expand Sections** button (≣) and choose as many of these options as required—**Equation**, **Override and Contribution**, **Discretization**, **Stabilization**, and **Advanced Physics Options**.

From either menu, select:

- **Reset to Default** to reset the sections to show only the **Equation Sections** and **Override and Contribution**. Or click the **Expand Sections** button to reset it so all sections are collapsed (not expanded).
- Select **Reset from Preferences** to reset the sections to display the system preferences defined from The Preferences Dialog Box.
- **Set as Preferences** to transfer the local settings made here to be used globally in all models. In other words, to make these selections the default system preferences as defined in The Preferences Dialog Box.

> These options are also accessed (and activated) from The Preferences Dialog Box in the **Model Builder** section.

SHOW MORE PHYSICS OPTIONS

There are several general options available for the physics interfaces and for individual nodes. This section is a short overview of these options, and includes links to additional information.

To display additional options for the physics interfaces and other parts of the model tree, click the **Show** button (👁) on the **Model Builder** and then select the applicable option.

After clicking the **Show** button (👁), additional sections are displayed on the settings window when a node is clicked and additional nodes are made available.

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

> In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

The additional sections that can be displayed include **Equation**, **Advanced Settings**, **Discretization**, **Consistent Stabilization**, and **Inconsistent Stabilization**.

You can also click the **Expand Sections** button (≣) in the **Model Builder** to always show some sections or click the **Show** button (👁) and select **Reset to Default** to reset to display only the **Equation** and **Override and Contribution** sections.

For most nodes, both the **Equation** and **Override and Contribution** sections are always available. Click the **Show** button (👁) and then select **Equation View** to display the **Equation View** node under all nodes in the **Model Builder**.

Availability of each node, and whether it is described for a particular node, is based on the individual selected. For example, the **Discretization**, **Advanced Settings**, **Consistent Stabilization**, and **Inconsistent Stabilization** sections are often described individually throughout the documentation as there are unique settings.

SECTION	CROSS REFERENCE
Show More Options and Expand Sections	Advanced Physics Sections The Model Builder
Discretization	Show Discretization Discretization (Node)
Discretization—Splitting of complex variables	Compile Equations
Consistent and Inconsistent Stabilization	Stabilization Numerical Stabilization
Constraint Settings	Weak Constraints and Constraint Settings
Override and Contribution	Physics Exclusive and Contributing Node Types

OTHER COMMON SETTINGS

At the main level, some of the common settings found (in addition to the Show options) are the **Interface Identifier**, **Domain Selection**, **Boundary Selection**, **Edge Selection**, **Point Selection**, and **Dependent Variables**.

At the node level, some of the common settings found (in addition to the Show options) are **Domain Selection**, **Boundary Selection**, **Edge Selection**, **Point Selection**, **Material Type**, **Coordinate System Selection**, and **Model Inputs**. Other sections are common based on application area and are not included here.

SECTION	CROSS REFERENCE
Coordinate System Selection	Coordinate Systems
Domain, Boundary, Edge, and Point Selection (geometric entity selection)	About Geometric Entities About Selecting Geometric Entities The Geometry Entity Selection Sections
Equation	Physics Nodes—Equation Section
Interface Identifier	Predefined and Built-In Variables Variable Naming Convention and Namespace Viewing Node Names, Identifiers, Types, and Tags
Material Type	Materials
Model Inputs	About Materials and Material Properties Selecting Physics Model Inputs and Multiphysics Couplings
Pair Selection	Identity and Contact Pairs Continuity on Interior Boundaries

Showing and Expanding the Equation Sections and Equation Node

To always display the **Equations** section in its expanded view on physics node settings windows, click the **Expand Sections** button (≣) and select **Equations**.

To display the **Equation** section on all physics nodes' settings windows, click the **Show** button (👁) and select **Equation Sections**.

To display the **Equation View** node under all physics nodes in the **Model Builder**, click the **Show** button ( ) and select **Equation View**.

These options are also accessed (and activated) from The Preferences Dialog Box in the **Model Builder** section.

- Equation View
- Physics Nodes—Equation Section

Showing Advanced Physics Options

To enable this feature, click the **Show** button ( ) and select **Advanced Physics Options**. This displays the **Advanced** section on physics settings windows. It also activates additional options on the context menus or toolbars.

ADVANCED SETTINGS

This section does not normally show unless the physics interface contains some advanced options.

CONSTRAINT SETTINGS

This section controls how constraints are enforced, usually in boundary conditions.

To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak form implementation.

ADDITIONAL ADVANCED PHYSICS OPTIONS

There are additional nodes that can be added to the model either from the context menu (from the **More** or **Global** submenus) or from the **Physics** toolbar.

- **Weak Contribution.** See Weak Contribution (PDEs) and Weak Contribution (ODEs and DAEs).
- Weak Constraint
- Pointwise Constraint
- **Weak Contribution on Mesh Boundaries.** This feature is similar to Weak Contribution but is active on mesh boundaries. See Weak Contribution (PDEs) and Weak Contribution (ODEs and DAEs).
- Global Equations**.** Also see Adding ODEs, DAEs, and Other Global Equations.
- Global Constraint. Also see Symmetric and Nonsymmetric Constraints and Constraint.
- **Weak Contribution**. See Weak Contribution (PDEs) and Weak Contribution (ODEs and DAEs).
- **Discretization**. See Discretization (Node) and Discretization Section Shape Function Types and Element Orders.

These settings can also be activated from The Preferences Dialog Box.

Show Discretization

There are two categories of discretization—a section on the physics interface node's settings window (described here) and adding a Discretization (Node) for global equation-based modeling.

To enable this setting, click the **Show** button ( ) and select **Discretization**. The options available for each node are described individually in the documentation.

DISCRETIZATION

Element Order and Shape Function Type

The **Element order** (or, more precisely, the order of the shape function) directly affects the number of degrees of freedom in the solution and the accuracy of the solution. Increasing the order of the elements roughly corresponds to a uniform mesh refinement. Most physics interfaces uses Lagrange elements, which can be of order 1 to 4 (in 3D) or 5 (in 1D and 2D), with 2 being the default order. You can change the order using the **Element order** list. The software adapts the order of the numerical integration to the element orders for the physics in the model. Some physics use special element types or a reduced element order for some of the field variables.

Some additional information is included in Discretization Section Shape Function Types and Element Orders and Finite Elements (described in the *COMSOL API Reference Manual*).

Discretization of Fluids

The following is an example of the choices of element order for the **Laminar Flow** interface:

- **P1+P1** means linear elements for both the velocity components and the pressure field. Linear elements are computationally cheaper than higher-order elements and are also less prone to introducing spurious oscillations, thereby improving the numerical robustness.
- **P2+P1** means second-order elements for the velocity components and linear elements for the pressure field. Second-order elements work well for low flow velocities.
- **P3+P2** means third-order elements for the velocity components and second-order elements for the pressure field. This can add additional accuracy but it also adds additional degrees of freedom compared to P2+P1 elements.

The abbreviation P_mP_n is often used to indicate the polynomial order of, in this case, the shape functions (elements) for the velocity components (m) and the pressure (n) when using tetrahedral or triangular elements. Here a corresponding nomenclature is used for all element shapes.

The theory about this is in P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley & Sons, 2000.

Accurate Boundary Fluxes

Some physics can create and compute variables that accurately represent the flux across all boundaries. To enable these variables, select the **Compute boundary fluxes** check box. Optionally, a smoothing can be applied when computing the boundary flux variables. You add smoothing by selecting the **Apply smoothing to boundary fluxes** check box.

Complex-Valued Variables

Under **Value types when using splitting of complex variables**, you can specify the **Value type** (**Real** or **Complex)** of dependent variables when **Split complex variables in real and imaginary parts** setting is activated in the Compile Equations node of any solver sequence used. The default is the complex value type, but you can specify that the value of a dependent variable is real to make sure that it does not get affected by small imaginary contributions, which can occur, for example, when combining a Time Dependent or Stationary study with a frequency-domain study. If the split complex variables setting is not active the value type is ignored.

For information about how to specify the splitting of complex variables, see Compile Equations.

Selecting Other Options from the Show Menu

Click the **Show** button ( ) and select one of the following options to enable nodes either in context menus or on the toolbars.

ADVANCED STUDY OPTIONS

Select **Advanced Study Options** to enable these options from the context menu, which can then be added to the **Model Builder**—Job Configurations and Solver Configurations. These nodes also display if they contain content.

- Right-click the **Study** node to enable Cluster Computing, **Cluster Sweep**, and **Batch Sweep** and Batch.
- Right-click the **Solver Configurations** node to be able to select the option Create a Custom Solver and add a **Solver** node without any added solver settings or other nodes.
- Right-click any of the **Study Step** nodes to enable the Multigrid Level option from the context menu.

ADVANCED RESULTS OPTIONS

To display the **Views** node under **Results**, select **Advanced Results Options**.

- User-Defined Views
- Results Analysis and Plots

STABILIZATION

Select **Stabilization** to display the **Consistent Stabilization** and **Inconsistent Stabilization** sections on the settings windows. If you have access to the **Level Set** and **Mixture Model** interfaces, it displays a **Stabilization** section instead.

For detailed information about this feature, see Numerical Stabilization and Consistent Stabilization and Inconsistent Stabilization Sections on Settings Windows.

OVERRIDE AND CONTRIBUTION

Select **Override and Contribution** to include the section in all physics nodes (for material models, sources, boundary conditions, and so on). For a specific node, the **Override and Contribution** section in its settings window contains lists of other nodes that the node is overridden by, other nodes that the node overrides, and other nodes that contributes with the node (to the total load or flux, for example).

- Physics Exclusive and Contributing Node Types
- Listing Overrides and Contributions
- Overridden Selections

5

Definitions

This chapter describes the available functionality in the Global Definitions and Definitions branches such as parameters, variables, functions, and coordinate systems.

About Global and Local Definitions

Depending on the geometric scope, there are two types of definitions nodes that can be defined in the Model Builder—**Global Definitions** and local **Definitions**.

Global Definitions

The Global Definitions main branch has features that apply to the entire model. Add the following features either using the toolbars (the **Home** toolbar for Windows users and the **Main** toolbar for Mac and Linux users) or right-click **Global Definitions** ( ) and choose an option from the context menu:

- Variables (a=): user-defined variables that can be used anywhere to simplify the specifications of some properties.
- Parameters (Pi): user-defined global, scalar values that are used to parameterize any part of the model. See Parameters.
- Functions (f(x)): function templates for creating user-defined functions based on analytic expressions or imported data, or specifying parameters for common function types such as step functions, ramps, and random functions.
- Geometry Subsequence ( ): 3D, 2D and 1D parametrized geometry subsequences which can be called as a subroutine from a geometry sequences in a Component. These features are only available from the context menu.
- **Groups** ( ): Add **Load Group** ( ) and **Constraint Group** ( ) nodes for use in load cases. These features are only available from the context menu. See Using Load Cases.

Definitions

Just as the Global Definitions branch collects user-defined parameters, variables, and functions accessible at all levels in the Model Builder, the *Definitions* branch (one per Component) collects the definitions of variables, functions, and other objects where the geometric scope is restricted to a single component.

Definitions are under the Component branch because several models can separately be defined in one multiphysics file, for example, when treating certain parts of the whole model in 2D and other parts in 3D. Definitions which in some way refer to domains in a geometry, the geometry's dimension or its coordinate names must therefore be held apart in different Component branches.

An example of the type of objects you can add under the Definitions branch is a *Selection* node ( ), which saves selections of geometric entities (boundaries, for example) that relate to a region or part of the overall geometry for reuse in operations later in the modeling process.

Add a **Component** node to the Model Builder, then add definitions with a local scope that apply to that specific model Component. Click or select features either using the **Definitions** toolbar or right-click **Definitions** ( ) and choose an option from the context menu:

Select among the following definition types:

- **Variables** (a=): add user-defined variables to simplify the expressions of other variables and properties. See Variables.
- **View** ( ): create a user-defined view to visualize the model. See User-Defined Views in the Visualization and Selection Tools chapter.

- **Mass Properties** ( ): set up nodes which compute quantities such as volume, mass, center of mass, and moment of inertia. See Mass Properties.
- **Functions** (f(x)): add user-defined functions based on analytic expressions or imported data, or specifying parameters for common function types such as step functions, ramps, and random functions. See Functions.
- **Probes** ( ): add a probe to monitor the development of a scalar-valued quantity (real or complex-valued number) during a dynamic simulation. See Probes.
- **Component Couplings** ( ): add nonlocal couplings inside or between Components, in the form of integration, mapping, projection and similar user-defined operators. See Component Couplings.
- **Selections** ( ): create a user-defined set of geometric entities for reuse throughout the model. See Named Selections in the Visualization and Selection Tools chapter.
- **Pairs**: There are two types of pairs:
  - **Contact Pair** ( ), which specifies two boundary selections that cannot penetrate each other under deformation. The contact pairs are only available if your license includes the Structural Mechanics Module or the MEMS Module.
  - **Identity Pair** ( ), which specifies two boundary selections (also available for edges and points) that coincide while belonging to different parts of an assembly. Special boundary conditions connect the physics in the two parts.

  Pairs are only available and necessary when the model geometry is an assembly. See Identity and Contact Pairs.
- **Coordinate Systems** ( ): create coordinate systems for use in the physics interfaces. See Coordinate Systems.
- **Perfectly Matched Layer (PML)** ( ) and **Infinite Elements** ( ): surround your model by a perfectly matched layer or infinite elements, behaving as an unbounded extension of the modeling domain. See Infinite Element Domains and Perfectly Matched Layers.
- **Extra Dimensions**: attach extra dimensions to a selection in the base geometry of a model. This is a technology preview feature enabled from The Preferences Dialog Box. See Using Extra Dimensions for more information.

About Parameters, Variables, and Expressions

PARAMETERS AND VARIABLES

Parameters and *variables* are used to parameterize and organize your model. They are different in their definition and use.

Parameters

Parameters are user-defined constant scalars with a global scope that are available for use throughout the Model Builder tree. In particular, they can be used for parameterization in the Geometry, Mesh, and Study branches. Important uses include:

- Parameterizing geometric dimensions
- Parameterizing mesh element sizes
- Defining parametric sweeps

A *parameter expression* can contain: numbers, other parameters, mathematical constants, physical constants, functions of parameter expressions, unary operators, and binary operators. Parameters can have units. For example, a parameter can be defined as `(exp(-pi*i)+a)*c_const`, where `a` is another parameter, but it is often a scalar numerical value for use in a parametric sweep, where that value is updated during the sweep.

Variables

A *variable*'s expression can contain numbers, parameters, mathematical constants, physical constants, other variables, functions of variable expressions, unary operators, and binary operators. Variables can also depend on dependent variables (the solution) and their derivatives, and they can have units. For example, a variable can be defined as `pi*(R_tube^2-r_tube^2)`, where `R_tube` and `r_tube` are two other variables or parameters, for example defined as `10[mm]` and `25[mm]`, respectively.

> Variables cannot be used in the Geometry and Mesh branches, and only to a limited amount in the Study branch.

Variables can have global or local scope depending on where they are defined. A variable with local scope is limited to a geometric entity level within a model component: the entire component's geometry or selected domains, boundaries, edges (3D only), or points.

Variables can make a model easier to understand by introducing short and descriptive names for complicated expressions. Most variables for material properties, coordinates, and other quantities defined on the computational domain are "field variables"—that is, they are defined so that they can vary in space and time. For example, a built-in variable for the density in a Solid Mechanics interface, `solid.rho`, represents a density ρ as $\rho(x, y, z, t)$ in 3D. You can visualize it in a surface or volume plot, for example, but you cannot use it in an ODE or a global evaluation, even if it should happen to be defined as a constant value. In such a situation, use a component coupling that computes an average quantity or a point probe to obtain a quantity with a global evaluation scope.

NAMING CONVENTIONS, RESERVED NAMES, AND ERRORS

Parameter names and variable names are case sensitive and must begin with a lowercase or uppercase letter (a–z or A–Z). All other characters in the name must be a lowercase or uppercase letter, a number 0–9, or an underscore (_). Dots (.) are also allowed in names but are best avoided because they have a special meaning as delimiters in the namespace. It is possible that the first part of a variable name containing a dot is misinterpreted as a dot symbol for accessing a variable in the namespace.

It is good practice to use descriptive names that are different from the names of built-in functions and constants. Some fundamental built-in mathematical and numerical constants and built-in variables have reserved names; defining a variable using a reserved name is not recommended because it can cause unexpected results. If you use a variable name that is a reserved name (see Summary of Built-In Variables With Reserved Names), the name appears in orange and if you move the cursor to the name, a tooltip such as **j is a reserved name** appears. The following names are reserved: `eps`, `nan`, `NaN`, `inf`, `Inf`, `i`, `j`, and `pi`. Also, when used in a model, errors about duplicate variable names occur if you defined parameter names using names of built-in variables for the geometry, mesh, and physics (`h`, `dom`, and similar names of built-in variables as well as the names of dependent variables and spatial coordinates in the model).

If the expression contains a syntax error, it appears in red. Syntax errors can be due to illegal characters, mismatched parentheses, and other syntactic errors. The **Error** node (⊗), which occurs when trying to solve a model with a syntax error, typically contains information about the position and expression where the syntax error is located.

Parameters

To create a **Parameters** node (Pi):

- From the **Home** (Windows users) or **Main** (Mac and Linux users) toolbar click **Parameters**, or
- Right-click **Global Definitions** (≡) and select **Parameters**.

If the node has been created previously—there can only be one—you will find the **Parameters** node under the **Global Definitions** node. Parameters are useful in the following context:

- As parameters in dimensions for geometric primitives or other geometry operations

- As parameters for the mesh generators, for example specifying mesh size
- As parameters to control some aspects of the solution process
- To quickly evaluate a mathematical expression, including unit conversion
- In physics settings, expressions, and coupling operators
- In expressions when evaluating results

PARAMETERS

Enter values in the **Parameters** table to define parameters used throughout the whole model. In the **Parameters** section you can enter parameters manually or import them from a text file.

- In the **Parameters** table or the field under the table, enter a parameter **Name**.
- In the **Expression** column or field, enter a parameter expression that defines the parameter value—including a unit if desired.
- The **Value** column displays the value of the parameter in the base unit system.
- In the **Description** column or field, enter an optional description.

You can save the parameters to a text file to reuse in other models. Click the **Save to File** button ( ) and enter a **File name** in the **Save to File** dialog box, including the extension .txt. Click **Save** to store the parameters in a text file or in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®. The information is saved in space-separated columns in the same order as displayed on screen. When saving to Excel, an **Excel Save** dialog box appears where you can specify the sheet and range and whether to overwrite existing data, include a header, or use a separate column for units.

You can import or load data in files from a spreadsheet program, for example, with the **Load from File** button ( ) and the **Load from File** dialog box that appears. Data must be separated by spaces or tabs. If there is already data in the table, imported parameters are added after the last row. Move or edit rows as required. If the license includes LiveLink™ *for* Excel® you can also load parameters from a Microsoft Excel Workbook spreadsheet. Then an **Excel Load** dialog box appears where you can specify the sheet and range and whether to overwrite existing data. It is also possible to import from a spreadsheet containing a separate column for units.

Variables

Choose to add **Global Variables** or **Local Variables** from the **Home** (Windows users) or **Main** (Mac and Linux users) toolbar or right-click the **Global Definitions** or **Definitions** nodes to add a **Variables** (a=) node to the Model Builder.

Use this node to define expressions as user-defined *variables*. Global variables can be used in any context which accepts variable expressions, in all Components and on all geometric entities—provided that their expressions are also global expressions. In contrast, local variables have a specific geometric domain of definition. Such variables can only be used and evaluated in a specific Component, or on selected domains, boundaries, edges, or points.

Global variables are primarily useful for expressions involving parameters that do not depend on the geometry, such as time, or dependent variables in an ODE or algebraic equation. Whenever possible, define variables under **Definitions** in a Component to minimize the risk of variable name conflicts in the global namespace.

Which variables are available for evaluation in postprocessing is decided at the time a solution is created. This means that variables you define do not immediately show up as predefined quantities in results nodes or become available for use in expressions when postprocessing an existing solution. To access the new variables, you must solve the model or update an existing solution by right-clicking a **Study** node ( ) and selecting **Update Solution** ( ).

GEOMETRIC ENTITY SELECTION (LOCAL DEFINITIONS ONLY)

Select the geometric scope from the **Geometric entity level** list—**Entire component**, **Domain**, **Boundary**, **Edge** (3D only), or **Point**. For all levels except **Entire component**, you must also specify the variables domain of definition either by adding entities to a **Manual** selection or choosing **All boundaries**, for example, from the **Selection** list.

Variables defined in a Component but with **Geometric entity level** set to **Entire component** are in fact global; they can be used anywhere using their full name. For example, if you define variable `a` in Component 1 in this way, you can refer to it in another Component as `comp1.a`.

VARIABLES

In the **Variables** table or the fields under the table, enter variables by defining a variable name under **Name**, an expression that defined the variable under **Expression** (see About Parameters, Variables, and Expressions), and (optionally) a description that explains the variable under **Descriptions**. Alternately, you can import variable definitions from a text file.

The **Save to File** button (💾) saves variables to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®) for reuse in other models. The information is saved in space-separated columns in the same order as displayed on screen. When saving to Excel, an **Excel Save** dialog box appears where you can specify the sheet and range and whether to overwrite existing data, include a header, or use a separate column for units.

Using the **Load from File** button (📂), you can import or load data in text files created, for example, by a spreadsheet program. Data must be separated by spaces or tabs. If the license includes LiveLink™ *for* Excel® you can also load variables from a Microsoft Excel Workbook spreadsheet. Then an **Excel Load** dialog box appears where you can specify the sheet and range, whether to overwrite existing data, and declare if the data is stored using a separate column for units.

For an example of global variables, see Effective Diffusivity in Porous Materials: model library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**

For examples of local variables see:

- Acoustics of a Muffler: model library path **COMSOL_Multiphysics/Acoustics/automotive_muffler**
- Tubular Reactor: model library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**
- Fluid Valve: model library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

Common Settings for the Definitions Nodes

Many nodes which can be added under **Global Definitions** or under **Definitions** in a Component share the same settings, or use settings generally found throughout COMSOL. Links to Common Settings Window Descriptions provides cross references to information relevant to some of these settings.

COMMON BUTTONS ON THE SETTINGS WINDOWS

The following buttons are available on many of the settings windows and are mostly self explanatory. These are not explicitly described or explained for every node.

- In general, use the **Move Up** (↑), **Move Down** (↓), and **Delete** (☒) buttons and the fields under tables to edit the table contents. Or right-click a table cell and select **Move Up**, **Move Down**, or **Delete**.
- The **Add** button (+) under a list of named selections opens an **Add** dialog box that contains all existing selections for the same geometric entity level.
- To save the contents of a table, click the **Save to File** button (💾) and enter a **File name** in the **Save to File** dialog box, including the extension `.txt`. Click to **Save** the text file. The information is saved in space-separated columns in the same order as displayed on screen.
- Use the **Load from File** button (📂) and **Load from File** dialog box to import data in text files, generated by, for example, a spreadsheet program. Data must be separated by spaces or tabs (or be in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®).

- About Selecting Geometric Entities
- The Model Builder
- Named Selections

IDENTIFIER AND NAME SETTINGS

Many nodes under **Global Definitions** or under **Definitions** in a Component contain a setting which requires a unique identifier or name.

Name

For Functions and component couplings you must specify a function name or operator name that is unique on the global level or within the model component where the feature is added. You can use this function name or operator name in expressions, and it is also the node's identifier. A unique default function name or operator name is always generated when the node is created.

Note that if you right-click the node in the Model Builder and select **Properties**, the name you entered appears in the **Identifier** field, while the Name field contains the descriptive name shown on the node in the Model Builder.

Identifier

Nodes that automatically generate variables generally have an identifier setting. The identifier must be unique on the global level or within the model component where the feature is added. It provides a namespace for variables created by the node. A unique default identifier is always generated when the node is created.

For example, coordinate system nodes define a variable that evaluates the determinant of the system's coordinate transformation matrix. Internally in the coordinate system, this variable's name is `detT`. If a coordinate system has the identifier `sys1`, the determinant variable can be accessed in equations and postprocessing as `sys1.detT`. A complete list of variables defined by a node is usually available in its **Equation View** subnode.

To display the **Equation View** node under all nodes creating variables, click the **Show** button and select **Equation View**. See also Equation View.

- The Node Settings Windows
- Editing Node Properties, Names, and Identifiers

LINKS TO COMMON SETTINGS WINDOW DESCRIPTIONS

- Name
- Plotting Functions
- Smoothing
- Geometry Subsequence

Geometry Subsequence

You can add a **Geometry Subsequence** node ( ) to the **Global Definitions** branch. It is a geometry sequence that can be called (several times) from other geometry sequences. You can use geometry subsequences to make your own parametrized geometric primitives. When calling a subsequence, the input to the subsequence is numerical values of its arguments. The output from the subsequence is a number of geometry objects and a number of selections on these objects. The geometry subsequence node behaves like an ordinary geometry sequence with a few minor differences as described in the Using Geometry Subsequences section.

See Creating a Geometry Sequence in the Geometry Modeling and CAD Tools chapter, which also explains The Geometry Node and the geometry features.

Operators, Functions, and Constants

Many built-in mathematical and logical operators, functions, and constants can be used to specify parameters, variables, equation coefficients, and material properties. These tables list the unary and binary operators (Table 5-1 and Table 5-2), special operators (Table 5-8), mathematical functions and constants (Table 5-6), and physical constants predefined as variables (Table 5-7) that are available in COMSOL Multiphysics. See also Component Couplings for information about coupling operators.

Unary, Binary, and List Operators and Their Precedence Rules

TABLE 5-1: UNARY OPERATORS

OPERATOR	DESCRIPTION
+	Unary plus
-	Unary minus
!	Logical not

The binary operators include arithmetic and logical operations.

TABLE 5-2: BINARY OPERATORS

OPERATOR	DESCRIPTION
+	Plus
-	Minus
*	Multiply
/	Divide
^	Power
==	Equal
!=	Not equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
&&	Logical and
||	Logical or

The following operators are used for precedence, grouping, lists, and unit definitions:

TABLE 5-3: GROUPING, LIST, AND UNIT OPERATORS

OPERATOR	DESCRIPTION
()	Parentheses for controlling precedence in expressions
{}	Vector and tensor expressions
,	Element separator in lists
.	Scoping operator
[]	Unit

The following list shows the precedence order for the operators above:

TABLE 5-4: PRECEDENCE LEVELS

PRECEDENCE LEVEL	SYMBOL	DESCRIPTION
1	`() {} .`	Grouping, lists, namespace
2	`^`	Power
3	`! - +`	Unary: logical not, minus, plus
4	`[]`	Unit
5	`* /`	Multiplication, division
6	`+ -`	Addition, subtraction
7	`< <= > >=`	Comparisons: less than, less than or equal, more than, more than or equal
8	`== !=`	Comparisons: equal, not equal
9	`&&`	Logical and
10	`||`	Logical or
11	`,`	Element separator in lists

Mathematical and Numerical Constants

The following table includes the built-in mathematical and numerical constants. The names of these constants are reserved names that you cannot use when creating user-defined variables and parameters.

TABLE 5-5: MATHEMATICAL AND NUMERICAL CONSTANTS

NAME	DESCRIPTION
`eps`	Floating point relative accuracy (machine epsilon, 2^{-52} or about $2.2204 \cdot 10^{-16}$, for double floating point numbers).
`i`, `j`	Imaginary unit, $\sqrt{-1}$.
`inf, Inf`	Infinity, ∞. A value larger than what can be handled with floating-point representation.
`NaN`, `nan`	Not-a-number. An undefined or unrepresentable value such as the result of `0/0` or `inf/inf`.
`pi`	Pi (about 3.141592653589793).

Mathematical Functions

The following list includes the built-in mathematical functions that you can use when defining variables or directly in expressions in the physics settings, for example. The function names are reserved names that cannot be used for user-defined functions, but they can be used for variable and parameter names.

TABLE 5-6: MATHEMATICAL FUNCTIONS

NAME	DESCRIPTION	SYNTAX EXAMPLE
`abs`	Absolute value	`abs(x)`
`acos`	Inverse cosine (in radians)	`acos(x)`
`acosh`	Inverse hyperbolic cosine	`acosh(x)`
`acot`	Inverse cotangent (in radians)	`acot(x)`
`acoth`	Inverse hyperbolic cotangent	`acoth(x)`
`acsc`	Inverse cosecant (in radians)	`acsc(x)`
`acsch`	Inverse hyperbolic cosecant	`acsch(x)`
`arg`	Phase angle (in radians)	`arg(x)`

TABLE 5-6: MATHEMATICAL FUNCTIONS

NAME	DESCRIPTION	SYNTAX EXAMPLE
`asec`	Inverse secant (in radians)	`asec(x)`
`asech`	Inverse hyperbolic secant	`asech(x)`
`asin`	Inverse sine (in radians)	`asin(x)`
`asinh`	Inverse hyperbolic sine	`asinh(x)`
`atan`	Inverse tangent (in radians)	`atan(x)`
`atan2`	Four-quadrant inverse tangent (in radians)	`atan2(y,x)`
`atanh`	Inverse hyperbolic tangent	`atanh(x)`
`besselj`	Bessel function of the first kind	`besselj(a,x)`
`bessely`	Bessel function of the second kind	`bessely(a,x)`
`besseli`	Modified Bessel function of the first kind	`besseli(a,x)`
`besselk`	Modified Bessel function of the second kind	`besselk(a,x)`
`ceil`	Nearest following integer	`ceil(x)`
`conj`	Complex conjugate	`conj(x)`
`cos`	Cosine	`cos(x)`
`cosh`	Hyperbolic cosine	`cosh(x)`
`cot`	Cotangent	`cot(x)`
`coth`	Hyperbolic cotangent	`coth(x)`
`csc`	Cosecant	`csc(x)`
`csch`	Hyperbolic cosecant	`csch(x)`
`erf`	Error function	`erf(x)`
`erfinv`	Inverse error function	`erfinv(x)`
`exp`	Exponential function e^x. That is, `exp(1)` is the mathematical constant e (Euler's number).	`exp(x)`
`floor`	Nearest previous integer	`floor(x)`
`gamma`	Gamma function	`gamma(x)`
`imag`	Imaginary part	`imag(u)`
`log`	Natural logarithm	`log(x)`
`log10`	Common logarithm (base 10)	`log10(x)`
`log2`	Base-2 logarithm	`log2(x)`
`max`	Maximum of two arguments	`max(a,b)`
`min`	Minimum of two arguments	`min(a,b)`
`mod`	Modulo operator	`mod(a,b)`
`psi`	Psi function and its derivatives	`psi(x,k)`
`range`	Create a range of numbers	`range(a,step,b)`
`real`	Real part	`real(u)`
`round`	Round to closest integer	`round(x)`
`sec`	Secant	`sec(x)`
`sech`	Hyperbolic secant	`sech(x)`
`sign`	Signum function	`sign(u)`
`sin`	Sine	`sin(x)`
`sinh`	Hyperbolic sine	`sinh(x)`

TABLE 5-6: MATHEMATICAL FUNCTIONS

NAME	DESCRIPTION	SYNTAX EXAMPLE
sqrt	Square root	sqrt(x)
tan	Tangent	tan(x)
tanh	Hyperbolic tangent	tanh(x)

Physical Constants

Physical constants are fundamental, universal constants that represent physical quantities. COMSOL Multiphysics includes the most widely used physical constants as built-in constants. Table 5-7 lists all supported physical constants with their names, symbol (variable name), value, and SI unit. The values are taken from Ref. 1 and include the SI unit.

TABLE 5-7: PHYSICAL CONSTANTS

NAME	SYMBOL	VALUE
Acceleration of gravity	g_const	9.80665[m/s^2]
Avogadro constant	N_A_const	6.02214129e23[1/mol]
Boltzmann constant	k_B_const	1.3806488e-23[J/K]
Characteristic impedance of vacuum (impedance of free space)	Z0_const	376.730313461...[ohm] (mu0*c)
Electron mass	me_const	9.10938291e-31[kg]
Elementary charge	e_const	1.602176565e-19[C]
Faraday constant	F_const	96485.3365[C/mol]
Fine-structure constant	alpha_const	7.2973525698e-3
Gravitational constant	G_const	6.67384e-11[m^3/(kg*s^2)]
Molar volume of ideal gas (at 273.15 K and 1 atm)	V_m_const	22.413968e-3[m^3/mol]
Neutron mass	mn_const	1.674927351e-27[kg]
Permeability of vacuum (magnetic constant)	mu0_const	4*pi*1e-7[H/m]
Permittivity of vacuum (electric constant)	epsilon0_const	8.854187817e-12[F/m]
Planck's constant	h_const	6.62606957e-34[J*s]
Planck's constant over 2 pi	hbar_const	1.054571726e-34[J*s]
Proton mass	mp_const	1.672621777e-27[kg]
Speed of light in vacuum	c_const	299792458[m/s]
Stefan-Boltzmann constant	sigma_const	5.670373e-8[W/(m^2*K^4)]
Universal gas constant	R_const	8.3144621[J/(mol*K)]
Wien displacement law constant	b_const	2.8977721e-3[m*K]

REFERENCE

1. *The NIST Reference on Constants, Units, and Uncertainty*, http://physics.nist.gov/cuu/Constants/index.html

Built-In Operators

There are special built-in operators available for modeling and for evaluating results; these operators are similar to functions but behave differently. Many physics interfaces use these operators to implement equations and special functionality. See Table 5-8 and the detailed descriptions that follow.

TABLE 5-8: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
`adj(`*`expr`*`)`	Evaluate expression using the adjoint sensitivity.	adj
`at`	Access the solution at any time.	at

`ballint(`*`r`*`,`*`expr`*`),`
`ballavg(`*`r`*`,`*`expr`*`),`
`circint(`*`r`*`,`*`expr`*`),`
`circavg(`*`r`*`,`*`expr`*`),`
`diskint(`*`r`*`,`*`expr`*`),`
`diskavg(`*`r`*`,`*`expr`*`),`
`sphint(`*`r`*`,`*`expr`*`),`
`sphavg(`*`r`*`,`*`expr`*`)`
	Evaluates the integral or average of the expression on the specified shape with radius *r*.	ball, circle, disk, and sphere

`bdf(`*`expr`*`,`*`i`*`)`	Apply backward differentiation formula of order *`i`* on expression.	bdf
`bndenv(`*`expr`*`)`	Evaluates the expression *`expr`* at the coordinates of a particle at a boundary.	env and bndenv
`centroid(`*`expr`*`)`	For simplex meshes it evaluates the expression *`expr`* in the centroid of the mesh element to which the point belongs.	centroid
`circumcenter(`*`expr`*`)`	Evaluates the expression *`expr`* in the circumcenter of the mesh element to which the point belongs.	circumcenter
`d(f,x)`	Differentiation operator. Differentiation of `f` with respect to `x`.	Differentiation Operators: d, pd, and dtang

`depends(`*`expr`*`)`
`depends(`*`expr`*`,`*`var`*`)`
	True if expression *`expr`* depends on the solution or *`var`*, respectively.	depends

`dest(`*`expr`*`)`	Evaluate parts of an integration coupling expression on destination side.	dest
`down(`*`expr`*`)`	Evaluate expression as defined in adjacent downside.	up and down
`dtang(f,x)`	Tangential differentiation of an expression `f` defined on a boundary with respect to a spatial dimension `x`.	Differentiation Operators: d, pd, and dtang

`emetric(exprx,expry)`
`emetric(exprx,expry,`
`exprz)`
	The square of the length of the global vector (exprx, expry, exprz) computed in the mesh element's own metric.	emetric

`env(`*`expr`*`)`	Evaluates the expression *`expr`* at the coordinates of a particle in a domain.	env and bndenv
`error(`*`'string'`*`)`	Generates an error with error message *`string`*.	error
`fsens(`*`expr`*`)`	Evaluate expression using the functional sensitivity.	fsens

TABLE 5-8: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
`if(`*`cond`*`,`*`expr1`*`,`*`expr2`*`)`	Conditional expression evaluating the second or third argument depending on the value of the condition.	if
`integrate(`*`expr`*`,`*`var`*`, lower,upper)`	Evaluate integral of general expression with respect to an integration variable over a real interval specified by lower and upper limits.	integrate
`isdefined(`*`variable`*`)`	Returns one where the variable is defined and zero where it is not defined.	
`isinf(`*`expr`*`)`	True if expression evaluates to infinity.	isinf and isnan
`islinear(`*`expr`*`)`	True if expression is a linear function of the solution.	islinear
`isnan(`*`expr`*`)`	True if expression evaluates to NaN (not-a-number).	isinf and isnan

`jacdepends(`*`expr`*`)`
`jacdepends(`*`expr`*`,`*`var`*`)`
	True if the derivative of the expression *`expr`* with respect to the solution depends on the solution or *`var`*, respectively.	jacdepends

`lindev`	Evaluate an expression linearized at the linearization point (when a linearization point is stored in the solution).	lindev
`linper`	Marks a load term to be used in a Linear perturbation solver.	linper
`linpoint`	Access the linearization point (when a linearization point is stored in the solution).	linpoint
`linsol`	Access the standard solution (for example inside `linpoint` or `lintotal`).	linsol
`lintotal`	Access the sum of the linearization point and linear perturbation.	lintotal
`lintotalavg`	Evaluate average of `lintotal(`*`expr`*`)` over all phases.	lintotalavg
`lintotalpeak`	Evaluate maximum of `lintotal(`*`expr`*`)` over all phases.	lintotalpeak
`lintotalrms`	Evaluate RMS of `lintotal(`*`expr`*`)` over all phases.	lintotalrms
`linzero`	Evaluate expression with zero solution.	linzero
`mean(`*`expr`*`)`	Mean value of expression as evaluated on adjacent boundaries.	mean
`nojac(`*`expr`*`)`	No contribution to the Jacobian.	nojac
`pd(f,x)`	Differentiation operator. Differentiation of `f` with respect to `x`. No chain rule for dependent variables.	Differentiation Operators: d, pd, and dtang
`ppr`	Accurate derivative recovery.	ppr and pprint
`pprint`	Accurate derivative recovery within each domain group.	ppr and pprint
`prev(`*`expr`*`,`*`i`*`)`	Evaluate expression at the *`i`*th previous time step.	prev

TABLE 5-8: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
`reacf`	Accurate evaluation of reaction forces and fluxes.	reacf
`realdot(a,b)`	Treat complex numbers *a* and *b* as real 2-vectors and return their dot product.	realdot
`scope.ati(coordinate exprs,expr)`	Evaluates the expression `expr` at an `i`-dimensional entity in the point with coordinates given by the coordinate expressions `coordinate exprs`.	spatial at
`sens(expr,i)`	Evaluate expression using the forward sensitivity for the parameter given by the second argument.	sens
`shapeorder(variable)`	The element order used for discretization of a variable.	shapeorder
`side(entity,expr)`	Evaluate expression as defined in the adjacent entity.	side
`subst(expr, expr1_orig, , expr1_subst,...)`	Substitute variables in an expression with other variables or expressions.	subst
`sum(expr,index, lower,upper)`	Evaluate sum of general expression for all indices from lower to upper.	sum
`test(expr)`	Test function operator.	test
`timeint, timeavg`	Integrate or compute the average of a time-dependent expression over a time interval.	timeint and timeavg
`try_catch(tryExpr, catchExpr)`	Attempts to evaluate `tryExpr`, but if this fails for any point, `catchExpr` is evaluated instead.	try_catch
`up(expr)`	Evaluate expression as defined in adjacent upside.	up and down
`var(expr,fieldname1, fieldname2, ...)`	Variation operator.	var
`with`	Access any solution.	with

ADJ

- When you apply the adjoint sensitivity operator `adj` to an expression, COMSOL uses the adjoint sensitivity solution instead of the primal solution for the evaluation.
- The adjoint sensitivity solution is available for results when the sensitivity solver has been used with the adjoint sensitivity method, and for the dependent variables that have been solved for.

AT

- The `at` operator can access a solution to a time-dependent problem at any time. COMSOL provides the solution at that time using interpolation.
- The first input argument is the time. The second input argument is the expression that you want to evaluate using this solution. For example, `at(12.5,u)` is the solution at 12.5 s.
- The `at` operator can only be used during results evaluation, so it should not be used when setting up the model.

BALL, CIRCLE, DISK, AND SPHERE

- The `ballint`(*r*, *expr*) operator computes the volume integral of the expression *expr* in a ball with radius *r* around the point in which it is evaluated. The `ballint` operator can be evaluated on all entities in 3D.

- The `ballavg`(*r*, *expr*) operator is defined as `ballint`(*r*, *expr*)/`ballint`(*r*, 1).
- The `circint`(*r*, *expr*) operator computes the curve integral of the expression expr on a circle with radius *r* around the point in which it is evaluated. The `circint` operator can be evaluated on all entities in 2D and on edges in 3D, when used in 3D the integration is done on the circle in the normal plane to the edge.
- The `circavg`(*r*, expr) operator is defined as `circint`(*r*, expr)/`circint`(*r*, 1).
- The `diskint`(*r*, expr) is similar to the `circint` operator but calculates the surface integral on a disk instead.
- The `diskavg`(*r*, expr) operator is defined as `diskint`(*r*, expr)/`diskint`(*r*, 1).
- The `sphint`(*r*, expr) is similar to the `ballint` operator but computes the surface integral on a sphere instead.
- The `sphavg`(*r*, expr) operator is defined as `sphint`(*r*, expr)/`sphint`(*r*, 1).
- All of operators can be used with a third argument *N* that approximately specifies the number of integration points used—for example, `circint`(*r*, expr,100).
- To all operators you can add a suffix ("*_frameId*") that specifies the frame in which the integration is done—for example, `circint_spatial`(*r*, expr).

BDF

- Use the `bdf` operator to approximate time derivatives when the time discrete solver is used.
- The expression `bdf(`*expr*, *i*`)` results in a discretization of the time derivative of *expr* using a backward differentiation formula.
- The second argument, *i*, determines the order of accuracy of the discretization. Currently, first order and second order is available, so allowed values are $i = 1$ and $i = 2$. A second-order formula requires access to two previous time steps. Because this is not possible at the initial step, the evaluation at the initial step always uses the first-order formula.
- The `bdf` operator can be implemented using the `prev` operator. For example, obtain the first-order backward differentiation formula, also known as the backward Euler method, through `bdf(u,1) = (u-prev(u,1))/timestep`.

CENTROID

- The `centroid`(*expr*) operator evaluates the expression *expr* in the centroid of the mesh element to which the point belongs for simplex meshes.
- Note that the operator is context sensitive in the sense that it chooses the mesh element of the same dimension (a point often belongs to several different mesh elements) as the context in which the evaluation is performed.

CIRCUMCENTER

- The `circumcenter`(expr) operator evaluates the expression expr in the circumcenter of the mesh element to which the point belongs. This point is in general only well defined for simplices, but for other mesh elements a natural generalization is available.
- Note that the operator is context sensitive in the sense that it chooses the mesh element of the same dimension (a point often belongs to several different mesh elements) as the context in which the evaluation is performed.
- You can add a suffix ("*_frameId*") that specifies the frame in which the evaluation is done—for example, `circumcenter_spatial`(expr).

DIFFERENTIATION OPERATORS: D, PD, AND DTANG

- All differentiation operators (`d`, `pd`, and `dtang`) can be used both in model settings and in results evaluation.
- Use the `d` operator to differentiate a variable with respect to another variable. For example, `d(T,x)` means differentiation of *T* with respect to *x*. Some space derivatives are also available using predefined variables. For example, `uxx`, `d(ux,x)`, and `d(d(u,x),x)` are equivalent for a dependent variable `u` when evaluated in a domain. On a boundary, however, `d(u,x)` is 0, while `ux` is the average of the values from the adjacent domains. The

expression `d(E,TIME)` computes the reference time derivative of the expression `E`.

- The `pd` operator works in a similar way to the `d` operator. The main difference is that `pd(u,x)` is 0 rather than `ux` (no chain rule is applied for dependent variables).
- Use the `dtang` operator to compute derivatives in the tangential direction along a boundary. The `dtang` operator can be applied to expressions that are only defined on the boundary and therefore cannot be differentiated by the `d` operator. In a 3D model, `(dtang(f,x), dtang(f,y), dtang(f,z))` is a vector in the tangent plane of a boundary at the point where it is evaluated, and similarly in a 2D model, `(dtang(f,x), dtang(f,y))` is a vector in the tangent line of a boundary point. When evaluated in a domain, `dtang(f,x)` is the same as `d(f,x)`. The second argument of `dtang(f,x)` must be one of the spatial coordinates. Not all quantities have rules for evaluating tangential derivatives. Applying `dtang(f,x)` to an expression with no tangential derivative rule results in an error message.

Examples of Using the Differentiation Operators

The expressions `d(u^2,u)` and `pd(u^2,u)` both equal `2*u`—`d` also takes the spatial and time variables into account and treats their derivatives accordingly. In other words, if u is the dependent variable and x and t are the spatial coordinate and time, respectively, then `d(u+x,x)` equals `ux+1` (`ux` is the spatial derivative of u with respect to x), while `pd(u+x,x)` equals `1` because u is considered to be independent of anything but itself in the case of `pd`. Equivalently, `d(u,t)` equals `ut`, while `pd(u,t)` is zero.

If u is a dependent variable defined only on a boundary, `d(u,x)` is not defined, but the tangential derivative `dtang(u,x)` can be evaluated on the boundary. The tangential derivative obeys most of the common differentiation rules, such as the product rule and the chain rule. It is worth pointing out, however, that `dtang(x,x)` is not always equal to 1.

DEPENDS

- The `depends(`*`expr`*`)` operator returns 1 if the expression *`expr`* that it operates on depends on the solution; otherwise it returns 0.
- `depends(`*`expr`*`,` *`var`*`)` returns 1 if *`expr`* depends on *`var`*; otherwise it returns 0.

Use this operator to check user-defined expressions for dependency on the solution.

DEST

The `dest` (destination) operator is available for use in integration coupling expressions. This operator forces the expression that it operates on to be evaluated on the destination points instead of the source points. This means that the destination operator can be used to create convolution integrals and other integral transforms. For instance, integrating the expression `u/((dest(x)-x)^2+(dest(y)-y)^2)` gives the following function of x and y:

$$f(x,y) = \int \frac{u(x',y')}{(x-x')^2+(y-y')^2}dx'dy'$$

EMETRIC

The `emetric(`*`vector`*`)` returns the square of the length of a vector computed in the mesh element's own metric. In this metric, the edges of the mesh element typically have length 1, and vectors orthogonal to the element have length 0. The number of input arguments defining the vector should be equal to the space dimension.

ENV AND BNDENV

> These operators can only be evaluated on particles in a particle tracing simulation. Thus, the Particle Tracing Module is required.

- Evaluating `env`(`expr`) on a particle evaluates `expr` at the point in the domain where the particle is. When evaluating a variable `var` on a particle, if the variable is not defined on the particle it is automatically replaced by `env`(`var`). Therefore the `env` operator can often be omitted.
- Evaluating `bndenv`(`expr`) on a particle, evaluates `expr` at the point on the boundary where the particle is. If the particle is not on a boundary, the evaluation fails. Use this operator instead of `env` when evaluating expressions that are only defined on boundaries.

ERROR

- The `error`(`string`) operator generates an error with error message `string`.
- You can for instance use this operator to make assertions on how your solution should behave. If you write `if(`*`cond,expr,error('cond is false')`*`)` you will calculate *`expr`* when *`cond`* is true and get an error message including the text *`cond is false`*, when *`cond`* is false.

FSENS

- The functional sensitivity operator `fsens` evaluates the sensitivity of the current objective functional with respect to the control variable given as the single argument. Note that the argument must be a control variable name; other dependent variables or general expression are not allowed.
- The functional sensitivity is available for analysis when the sensitivity solver has been used with either the adjoint or the forward sensitivity method, and for the control variables that have been solved for.

IF

- The `if(`*`cond,expr1,expr2`*`)` operator implements a conditional expression.
- The first argument is a condition that COMSOL treats as a Boolean expression. If—at a particular evaluation point—`cond` is true, then the second argument is evaluated, otherwise the third argument is evaluated. That is, only one branch is evaluated at any evaluation point.
- Use the `if` operator to avoid illegal operations. For example, `if(x==0,1,sin(x)/x)` is a valid and continuous expression for all values of x, including $x = 0$.

INTEGRATE

- `integrate(`*`expr,var,lower,upper`*`)` computes the integral of *`expr`* for the integration variable *`var`* over an interval specified by expressions *`lower`* for the lower limit and *`upper`* for the upper limit. The expressions for lower and upper limits do not have to be constants but are required to evaluate to real values.
- `integrate(`*`expr,var,lower,upper,tol`*`)` sets the relative tolerance in the numerical integration to *`tol`*. The default value of the relative tolerance (used when the fifth argument is omitted) is `1e-3`. The tolerance must be a real constant between 0 and 1.

ISINF AND ISNAN

- The `isinf` operator returns 1 if the expression that it operates on evaluates to infinity (or minus infinity); otherwise it returns 0.
- The `isnan` operator returns 1 if the expression that it operates on evaluates to NaN (not-a-number); otherwise it returns 0.

ISLINEAR

- The `islinear` operator returns 1 if the expression that it operates on is a linear function of the solution; otherwise it returns 0.
- Use this operator to check user-defined expressions for linearity with respect to the solution. The stationary solver does this automatically to choose between a linear or a nonlinear solver.

JACDEPENDS

- The `jacdepends`(*`expr`*) operator returns 1 if the derivative of the expression *`expr`*, with respect to any part of the solution, depends on the solution; otherwise it returns 0.
- `jacdepends`(*`expr`*, *`var`*) returns 1 if the derivative of the expression *`expr`*, with respect to any part of the solution, depends on *`var`*; otherwise it returns 0.

LINDEV

The `lindev` operator evaluates its argument in the following way when the solution has a stored linearization point: The expression is first linearized at the linearization point and then evaluated at the current solution. In particular, if `f` depends linearly on the solution, `lindev(f)` is the same as `f`. If `f` does not depend on the solution, `lindev(f)` is 0. If the solution does not have a stored linearization point, using `lindev` causes an error message.

LINPER

The `linper` operator has one single use: To indicate load terms that should be used by stationary solvers setting **Linearity** to **Linear perturbation** in the **Stationary Solver** node's settings. Terms not enclosed by `linper` are ignored by such solvers. On the other hand, terms inside `linper` are ignored by all other solvers. Evaluating the `linper` operator always gives the result 0.

LINPOINT

The `linpoint` operator can access the linearization point in a solution with a stored linearization point. If the solution does not have a stored linearization point, using `linpoint` causes an error message.

LINSOL

The `linsol` operator evaluates an expression using the standard solution. This is the default, so in most contexts the operator is not very useful. However, it can for example be used inside the argument of `linpoint` to evaluate a part of the expression with the standard solution instead of the linearization point.

LINTOTAL

The `lintotal` operator can access the sum of the linearization point and the linear perturbation in a solution with a stored linearization point. If the solution does not have a stored linearization point, using `lintotal` causes an error message.

LINTOTALAVG

The `lintotalavg` operator evaluates the average of an expression over all phases. `lintotalavg(f)` is evaluated by taking the average of `lintotal(f)` with the solution (but not the linearization point) multiplied by $e^{i\varphi}$ for a number of phases φ. The number of phases is automatically selected to achieve an accurate value. `lintotalavg(f,n)` uses `n` equidistantly spaced phases.

LINTOTALPEAK

The `lintotalpeak` operator evaluates the maximum of an expression over all phases. `lintotalpeak(f)` is evaluated by taking the maximum of `real(lintotal(f))` with the solution (but not the linearization point) multiplied by $e^{i\varphi}$ for a number of phases φ. The number of phases is automatically selected to achieve an accurate value. `lintotalpeak(f,n)` uses `n` equidistantly spaced phases.

LINTOTALRMS

The `lintotalrms` operator evaluates the RMS of an expression over all phases. `lintotalrms(f)` is the same as `sqrt(lintotalavg(abs(f)^2))`.

LINZERO

The `linzero` operator evaluates an expression using a zero solution. This is mostly used internally in the definitions of some other operators.

MEAN

- COMSOL can evaluate expressions on entities of different dimensions and this might affect the result. For instance, a point can lie on an edge, which can be an edge of a square, which can be part of the boundary of a cube. If you now want to access the numbering of the entity it is obvious that you should get different results for the point, edge, square and cube. If on the other hand you want to know the value of the dependent variable, this should be the same if you think of the point as part of a point, edge, square, or cube.
- The `mean` operator can be called on any entity that has a lower dimension than the model. The dimension of the entity from where the call is made is called `n`.
- The `mean` operator determines the smallest integer `m>n` for which there are adjacent entities of dimension `m.` It then evaluates the expression at the point one time for each `m` dimensional adjacent entity, regarding the point as a point in the entity, and takes the average of the calculated values.

See also up and down and side for similar operators.

NOJAC

- The `nojac` operator makes sure that any expression that it operates on is excluded from the Jacobian computation. This is useful if a Jacobian contribution is not strictly necessary and the computational requirements for it are high, such as when using a nonlocal coupling. The use of the `nojac` operator can then significantly lower the memory requirements by avoiding fill-in of the Jacobian matrix, but its use might also slow down the convergence of the solution.
- The *k*-ε turbulence model is an example where a built-in use of the `nojac` operator improves performance.

PPR AND PPRINT

- When the `ppr` operator is applied on an expression, COMSOL uses *polynomial-preserving recovery* to evaluate all variables in the expression that are discretized using Lagrange shape functions. For example, if `e = ux + vy`, then `ppr(e^2)=(ppr(ux)+ppr(vy))^2`.

Accurate Derivative Recovery

- The `pprint` operator similarly applies polynomial-preserving recovery within each group of domains with equal settings. Use these operators to get an estimate of the discretization error in the gradient. For example, `ux-pprint(ux)` in a 1D model.

If these operators are applied on expressions that are used when solving the model, COMSOL computes the Jacobian approximately by ignoring the operator. For example, the Jacobian of `ux-pprint(ux)` is 0.

PREV

When the time discrete solver is used, it stores the solution at a number of previous time steps.

- The expression `prev(`*`expr`*`,`*`i`*`)` evaluates *`expr`* using the solution obtained i time steps before the current time step.
- The operator can be used in equations as well as for results evaluation.
- When used in equations, the `prev` operator makes it possible to discretize time derivatives. For example, to discretize `ut` (the time derivative of u) with the formula known as the *backward Euler method*, use the expression `(u-prev(u,1))/`*`timestep`*. Here, **`timestep`** is the size of the time step used to reach the current solution `u`. The `prev` operator is also applicable for *`timestep`*. For example, `prev(`*`timestep`*`,1)` is the size of the time step used to reach the solution at the previous time step.
- When using the `prev` operator, sufficiently many previous time steps must be stored. Specify the number of previous time steps to store in the time discrete solver (time discrete levels) in the **Number of time discrete levels** field in the **General** section of the **Time Discrete Solver** node's settings window. Evaluating an expression at a previous time step that has not been stored results in an error.

REACF

The reaction force operator (`reacf`) evaluates the reaction force at each node point where a constraint is applied. The reaction force at a node is equal to the corresponding component of the negated residual vector $-L$ computed while solving the model. The reaction forces are stored together with the solution vector by the solvers.

- The reaction force operator (`reacf`) is useful when calculating integrals of reaction forces or fluxes.
- Apply the `reacf` operator on the names of dependent variables when doing a surface integration. For example, in structural mechanics, with dependent variables u and v corresponding to x- and y-displacements, use `reacf(u)` and `reacf(v)` to access integrals of the reaction forces in the x- and y-direction, respectively. The integration for the reaction force is a summation over the nodes, so the integration method must be summation rather than integration. The automatic integration method in the integrations available under **Results>Derived Values** detects the use of the `reacf` operation and then uses the summation method.
- Storing of the reaction forces can be disabled by clearing the **Reaction forces** check box in the **Output** section in the solver's settings window. This saves some computational time and memory. It is then not possible to use the reaction force operator.
- When using weak constraints, the residual vector is always 0, so reaction forces are not available.

REALDOT

- The expression `realdot(a,b)` treats complex numbers a and b as if they were real-valued vectors of length 2 and returns their dot product. Also think of the operator call as a shorthand form of `real(a*conj(b))`. This expression, however, is not an analytical function of its complex arguments and therefore has no unique partial derivatives with respect to a and b.
- The difference between `realdot(a,b)` and `real(a*conj(b))` is that the partial derivatives of the former with respect to a and b are defined as $\mathrm{conj}(b)$ and $\mathrm{conj}(a)$, respectively, while for the latter expression, the partial derivatives are $\mathrm{real}(\mathrm{conj}(a))$ and $\mathrm{real}(a)$.

The difference between the partial derivative definitions is important during sensitivity analysis of frequency-response problems (scalar or vector Helmholtz equations).

- Common objective function quantities like power and energy must be redefined in terms of `realdot(a,b)` rather than `real(a*conj(b))` for the sensitivity solver to compute correct derivatives. This applies also to the absolute value, `abs(a)`, via the definition $|a|^2 = \mathrm{realdot}(a, a)$.

SENS

- When the forward sensitivity operator (`sens`) is applied to an expression, COMSOL uses the forward sensitivity solution with respect to the indicated control variable instead of the primal solution for the evaluation. This means, in practice, that when the first argument is a linear expression in the dependent variables, the operator returns its derivative with respect to the control variable given as second argument. The result for a nonlinear expression usually lacks meaning.
- The forward sensitivity solution is available for analysis when the sensitivity solver has been used with the forward sensitivity method, and for the dependent variables and control variables that have been solved for.
- For scalar control variables, access the corresponding forward sensitivity solution by giving the control variable name as the second argument to this operator. For example, with the dependent variable u and the scalar control parameter q, access the forward sensitivity solution $\partial u/\partial q$ as `sens(u,q)`.
- For a control variable field, which is not a scalar, a more elaborate syntax specifying a unique degree of freedom must be used. This is done by giving an integer as the second argument, corresponding to the global degree of freedom number for the requested control variable degree of freedom.

SHAPEORDER

- The expression `shapeorder(`u`)` gives the element order used for discretization of the variable u.
- The argument u must be a dependent variable or a partial derivative of a dependent variable. In the latter case, the order returned is the order of the dependent variable itself and not the order of its derivative.
- It is an error to apply the `shapeorder` operator to, for example, an expression, a constant, or a spatial coordinate.

SIDE

- COMSOL can evaluate expressions on entities of different dimensions and this might not affect the result. For instance, a point can lie on an edge, which can be an edge of a square, which can be part of the boundary of a cube. If you now want to access the numbering of the entity it is obvious that you should get different results for the point, edge, square, and cube. If you, on the other hand, want to know the value of the dependent variable, this should be the same if you think of the point as part of a point, edge, square, or cube.
- The `side` operator can be called on an entity that has lower dimension than the model. The dimension of the entity from where the call is made is called `n`.
- The `side` operator is an operator that evaluates an expression, not on the entity where it is called but instead on one of the adjacent entities of dimension `n+1`. You choose which entity by giving its number (this is the number displayed, for instance, in the selection fields) as the first argument to the operator.
- It can happen that the entity you choose is adjacent to the evaluation point more than once. For instance, a boundary can have the same domain on both sides. In such cases the `side` operator takes the average of the different values.

See also up and down and mean for similar operators.

SPATIAL AT

- The *`scope`*`.at`*`i`*`(`*`<coordinate expressions>`*`, `*`expr`*`)` operator evaluates the expression *`expr`* in the geometry with the given model scope on an i-dimensional entity in the point given by the coordinate expressions. For example, `root.comp1.at1(0,y,dom)` evaluates `dom` in the two-dimensional geometry on an edge in the point $(0, y)$.
- To all operators you can add a suffix ("_frameId") that specifies the frame in which the coordinate expressions are used, for example, `at1_spatial(`x,y, *`expr`*`)`.

SUBST

- The `subst` operator takes a variable or expression as its first argument, followed by one or more argument pairs, each consisting of a variable name and an expression. The first argument in each pair is an original variable that appears in the variable or expression that you specify as the first argument, and the second argument in each pair is the variable or expression that you want to substitute the original variable with. This can be useful, for example, for replacing the variable for temperature in a temperature-dependent expression for some quantity by a fixed initial temperature for use as an initial condition.

- As an example, the expression `subst(hmnf.nutildeinit,p,pin_stat)` (taken from the Sajben Diffuser model in the CFD Module model library) substitutes the dependent variable for pressure, `p`, with a user-defined variable `pin_stat` for the inlet static pressure. The evaluation of the variable `hmnf.nutildeinit` (for the undamped turbulent kinematic viscosity) then takes the value of `pin_stat` instead of `p`.
- The unit of the output from the `subst` operator is the same as the unit for its first input argument.

SUM

The `sum` operator, when used as `sum(`*`expr`*`, `*`index`*`, `*`lower`*`, `*`upper`*`)`, is a summation operator that computes the sum of the terms *`expr`* for all *`index`* values from *`lower`* to *`upper`*. The expressions for lower and upper limits are required to evaluate to real values and to be independent of the evaluation point. For example, `sum(i^2,i,1,4)` = $1^2+2^2+3^2+4^2 = 30$.

TEST

- The `test` operator is available for modeling using the weak formulation of the equations.
- This operator creates the test function for the variable that it operates on. For an expression, $\mathrm{test}(F(u, \nabla u))$, the test operator is equivalent to:

$$\sum_i \mathrm{test}(u_i)\frac{\partial}{\partial u_i}F(u_i, \nabla u_i) + \mathrm{test}(\nabla u_i)\frac{\partial}{\partial \nabla u_i}F(u_i, \nabla u_i)$$

for all dependent variables u_i.

TIMEINT AND TIMEAVG

- The `timeint` and `timeavg` operators integrate and compute the average of a time-dependent expression over a time interval, respectively. `timeint(t1,t2,expr)` and `timeavg(t1,t2,`*`expr`*`)`compute the integral and average of *`expr`* over the interval `t=t1` to `t=t2`, respectively. The first two arguments must be real scalars. The integral is computed by numerical integration, subdividing the interval until the required accuracy is reached. The `timeavg` operator numerically integrates the expression in the same way as `timeint` and then divides the result by `t2-t1`.
- `timeint(t1,t2,`*`expr`*`,`*`tol`*`)` and `timeavg(t1,t2,`*`expr`*`,`*`tol`*`)` set the relative tolerance in the numerical integration procedure to *`tol`*. The tolerance must be a positive real scalar. The default tolerance (used when the fourth argument is omitted) is `1e-8`.
- `timeint(t1,t2,`*`expr`*`,`*`tol`*`,`*`minlen`*`)` and `timeavg(t1,t2,`*`expr`*`,`*`tol`*`,`*`minlen`*`)` set the smallest length of the subintervals used in numerical integration as a fraction of the length of the whole integration interval. Subintervals smaller than this length are not further subdivided even if that means that the required accuracy is not reached. *`minlen`* must be a positive real scalar. The default value of *`minlen`* (used when the last argument is omitted) is `1e-6`.
- The `timeint` and `timeavg` operators can only be used during results evaluation, so they should not be used when setting up the model.

TRY_CATCH

- The `try_catch(`*`tryExpr`*`,`*`catchExpr`*`)` operator attempts to evaluate the expression *`tryExpr`*, but if this fails for any point, the operator evaluates *`catchExpr`* instead.
- Note that the result might depend on how the mesh elements are partitioned into blocks during evaluation, which can be rather arbitrary. As soon as evaluation of the first argument fails in some part of the block, the

second argument gets evaluated in the entire block. Also, during postprocessing the behavior might change because NaN (Not-a-Number) values in a subset of the evaluation points are then accepted, so then the second argument's expression might not be evaluated even at points where the first argument fails.

- Consider using the `if` operator (see if) with a suitable condition as the first argument instead of the `try_catch` operator, if the `if` operator is applicable.

UP AND DOWN

- COMSOL can evaluate expressions on both sides of a boundary. One way to do this is by using the `up` and `down` operators. These operators are available only on boundaries (that is, geometric entities of dimension one less then the dimension of the model).
- For an expression or a variable that is discontinuous across a boundary, the value is different on either side, and COMSOL normally displays the mean values on the boundary.
- Use the `up` and `down` operators to evaluate an expression on the upside or downside of the boundary. If the upside or downside is outside of the geometry, or if the variables in the expression are not active on that side, the `up` or `down` operator returns `0`.

For more information about the upside and downside of a boundary, see Tangent and Normal Variables. See also side and mean for similar operators.

VAR

- The `var` operator (variation operator) is available for modeling using the weak formulation of the equations.
- The `var` operator has the same function as the `test` operator but is limited to the specified set of fields.
- This operator creates the test function for the variable that it operates on. For an expression, var(F(u, ∇u, v, ∇v), a), where the dependent variable u is in the field named a and the dependent variable v is not, the `var` operator is equivalent to:

$$\sum_i \mathrm{test}(u_i)\frac{\partial}{\partial u_i}F(u_i, \nabla u_i, v_i, \nabla v_i) + \mathrm{test}(\nabla u_i)\frac{\partial}{\partial \nabla u_i}F(u_i, \nabla u_i, v_i, \nabla v_i)$$

for all dependent variables u_i.

WITH

- The `with` operator can access any solution during results evaluation.
- For time-dependent problems, parametric problems, and eigenvalue problems, this makes it possible to use the solution at any of the time steps, any parameter value, or any eigensolution in an expression used for plotting or data evaluation.
- Use the solution number as the first input argument. The second input argument is the expression that you want to evaluate using this solution. For example, `with(3,u^2)` provides the square of the third eigensolution for an eigenvalue problem.
- For example, you can use the `with` operator to verify that two eigensolutions are orthogonal.
- The `with` operator can only be used during results evaluation, so it should not be used when setting up the model.

Predefined and Built-In Variables

This section provides information about available predefined and built-in variables that represent properties of the physics, geometry, mesh, and other parts of the model, including some tips on how you can use them in models.

Predefined Physics Variables

Physics variables are predefined variables that the physics introduce. They are typically functions of the dependent variables and their derivatives. Many of these variables are available in the **Predefined quantities** lists in the settings for plots and other results nodes.

To access physics variables, use a variable scoping syntax that uses the interface identifier to indicate the physics that they belong to.

The **Equation View** subnode is available for all physics nodes and contains a table with the names, expressions, units, and descriptions for the physics variables that the node defines. To display the **Equation View** subnodes, click the **Show** button ( ) and select **Equation View** from the **Model Builder**.

Variable Naming Convention and Namespace

COMSOL Multiphysics uses a namespace with a hierarchical structure to control the access to variables within a model component and variables in other components within the same model. To access variables, use the following namespace syntax:

- To refer to the top level of the model tree, use `root`.
- To refer to variables in a Component branch, use its *component identifier*, such as `comp1`.
- To refer to variables in a physics interface, use its *interface identifier*, such as `solid`.
- To refer to material properties, use the material node's tag, such as `mat1`, and the tag of the property group for the material property group, which is typically `def` for the Basic property group. For example, to access the density ρ in Material 1 use `mat1.def.rho` (or, using the full name for Component 1, `root.comp1.mat1.def.rho`; see below). Use this pattern when referencing to other material properties too.
- COMSOL evaluates the physics variables in the model component's namespace, so you need to prepend the interface identifier to access these variables. For example, `solid.disp` refers to the total displacement in a Solid Mechanics interface with the interface identifier `solid`.
- The dependent variables (field variables) are unique within a model, and you do not need the interface identifier to access them. For example, enter `T` to access the temperature in a Heat Transfer interface using the default name for the temperature.
- When referring to a variable you only need to provide the part of the full name that makes the variable unique. For example, within a Solid Mechanics interface `solid` in Component 1 `comp1`, it is sufficient to type `solid.disp`, but `comp1.solid.disp` and the full name `root.comp1.solid.disp` are also correct. To access the same variable from another Component, use `comp1.solid.disp` or `root.comp1.solid.disp`. The same mechanism applies to variables defined within a component. To access a global parameter `param1`, you can use `param1` directly or `root.param1`.

These variable naming conventions mean that the syntax becomes shorter when defining variables locally in a Component branch instead of globally. For example, to access the x-component of the electric field, $\boldsymbol{E}_x$, in an

Electrostatics interface with the interface identifier `es` in a Component 1 with the identifier `comp1`, you can use `es.Ex` in a variable defined in Component 1, but for a variable defined globally, the syntax is `comp1.es.Ex`.

Variable Classification and Geometric Scope

COMSOL provides a set of variables that you can use in expressions when specifying a model and for visualizing and analyzing the solution. A number of variables are common to all physics interfaces in a Component, for example, the spatial coordinate variables x, y, and z (for 3D and planar 1D and 2D geometries).

Every physics interface also has its own set of variables to represent quantities relevant to the physics or equations that it covers. Characteristics of variables include:

- Parameters and geometric variables are always available.
- The choice of physics interfaces and the dimension of the geometries in the model affect the set of available field variables and special variables.
- Equations can be active in different domains, which also affects the set of available variables. Variables corresponding to certain equation terms are available only in the particular part of the geometry (such as domains, boundaries, or points) where the equation is active.
- Variables defined on boundaries, edges, and points are active if the adjacent domain is active.

Variables are divided into the following general categories:

- User-defined *variables*
- Built-In Global Variables
- Physical Constants (predefined universal constants)
- Geometric Variables and Mesh Variables (variables that characterize geometric properties).
- *Field variables* (dependent variables and variables derived from them). For example, Shape Function Variables and Predefined Physics Variables.
- Component Couplings
- Solver Variables (available only during the solution process)

Built-In Global Variables

The following variables represent time (`t`), frequency (`freq`), eigenvalue (`lambda`), and the number of degrees of freedom (`numberofdofs`).

THE TIME VARIABLE

- For time-dependent problems use the time variable (`t`) with the unit seconds (s).
- It can be part of any expression in the point, edge, boundary, and domain settings, as well as during analysis.
- It is always scalar, even when the solution contains more than one output time.
- For stationary models, the value of `t` is 0.
- The value of `t` for results evaluation corresponds to the selection made in the **Time** list in the **Data** sections for the visualization and data evaluation nodes in the **Results** branch in the **Model Builder**. See the Results Analysis and Plots section.

THE FREQUENCY VARIABLE

The frequency variable (`freq`) is the global frequency for models in the frequency domain (time-harmonic models and frequency response analysis, for example).

THE EIGENVALUE VARIABLE

- When specifying an eigenvalue problem, use the eigenvalue variable (lambda) like any other variable.
- The eigenvalue solver extracts the Jacobian, the damping matrix, and the mass matrix through Taylor expansion of the problem with respect to the eigenvalue variable around a specified eigenvalue linearization point (which is zero by default).
- Other solvers treat the eigenvalue variable as a constant with value zero, unless it is set by an eigenvalue solution used as initial solution.
- After solving an eigenvalue problem, the eigenvalue name is available as a scalar variable for use in expressions.
- To choose between different eigenvalues, select one from the **Eigenvalue** list in the **Data** sections for the visualization and data evaluation nodes in the **Results** section of the **Model Builder**. The value of the eigenvalue variable corresponds to the selection made in the **Eigenvalue** list. See the Results Analysis and Plots section.
- For many physics interfaces, the default is to use an eigenfrequency study and compute and display the eigenfrequencies rather than the eigenvalues.

THE NUMBER OF DEGREES OF FREEDOM VARIABLE

The variable numberofdofs returns the total number of degrees of freedom (DOFs), which is the number of DOFs solved for plus any internal DOFs that the solver might add. The number of DOFs solved for plus the number of internal DOFs are reported in the **Messages** window when you compute the solution. COMSOL sometimes uses internal DOFs for storing information during solution that would be expensive or impossible to recompute afterward. Internal DOFs have no equations and therefore do not make the system matrices larger.

Geometric Variables and Mesh Variables

The variables that characterize geometric properties and the mesh are listed in Table 5-9, with detailed descriptions for some of the variables following the table.

TABLE 5-9: GEOMETRIC VARIABLES AND MESH VARIABLES

VARIABLE	DESCRIPTION
curv curv1,curv2	The curvature of a boundary in 2D is called curv. A boundary in 3D has two principal curvatures corresponding to the minimal and maximal normal curvatures. They are called curv1 and curv2, respectively. See Curvature Variables for details.
dom	The domain number, the boundary number, the edge number, or the vertex (point) number (all are integer values).
dvol	The volume scale factor variable, dvol, is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates. For 3D domains, this is the factor that the software multiplies volumes by when moving from local coordinates to global coordinates. In 2D and 1D domains, it is an area scaling factor and length scaling factor, respectively. If a moving mesh is used, dvol is the mesh element scale factor for the undeformed mesh. The corresponding factor for the deformed mesh is named dvol_spatial.
h	Available on all geometric entities, the variable h represents the mesh element size in the material/reference frame (that is, the length of the longest edge of the element).

TABLE 5-9: GEOMETRIC VARIABLES AND MESH VARIABLES

VARIABLE	DESCRIPTION
linearizedelem	In some calculations COMSOL forces mesh elements to become linear. This variable returns one inside such an element and zero otherwise. Note that the faces of the linearized mesh elements are not considered to be linearized themselves. You can use this variable to identify mesh elements with linearized elements.
meshtype	The mesh type index for the mesh element. This is the number of edges in the element.
meshelement	The mesh element number.
meshvol	Volume of the (linearized) mesh element.
n*x*, n*y*, n*z*	See Normal Variables.
qual	A mesh quality measure.
reldetjac reldetjacmin	The determinant of the Jacobian matrix for the mapping from the straight mesh element to the possibly curved element used when solving. Use this variable to measure the difference in shape between a curved element and the corresponding straight element. The variable reldetjacmin is a scalar for each element defined as the minimum value of the reldetjac variable for the corresponding element. A reldetjacmin value less than zero for an element means that the element is wrapped inside-out; that is, the element is an *inverted mesh element*.
s, s1, s2	See Parameterization Variables.
tcurv*x*, tcurv*y* (2D) tcurv1*x*, tcurv1*y*, tcurv1*z*, tcurv2*x*, tcurv2*y*, tcurv2*z* (3D)	Tangential directions for the corresponding curvatures. See Curvature Variables for more information.
t*x* and t*y* (2D) t1*x*, t1*y*, t1*z* (3D edges) t2*x*, t2*y*, t2*z* (3D surfaces)	See Tangent Variables.
x, y, z r, z	See Spatial Coordinate Variables.

When entering the spatial coordinate, parameterization, tangent, and normal geometric variables, replace the letters highlighted below in *italic font* with the actual names for the dependent variables (solution components) and independent variables (spatial coordinates) for the Component node.

For example, replace *u* with the names of the dependent variables in the model, and replace *x*, *y*, and *z* with the first, second, and third spatial coordinate variable, respectively. x_i represents the *i*th spatial coordinate variable. If the model contains a deformed mesh or the displacements control the spatial frame (in solid mechanics, for example), you can replace the symbols x, y, and z with either the spatial coordinates (`x`, `y`, and `z` by default) or the material (reference) coordinates (`X`, `Y`, and `Z` by default).

The variables `curv`, `dvol`, `h`, `qual`, `reldetjac`, and `reldetjacmin` are based on the mesh viewed in the material (reference) frame. If you have a moving mesh, the corresponding variables for the mesh viewed in the spatial frame have a suffix `_spatial` (that is, `curv_spatial`, `dvol_spatial`, and so on). If you use a deformed geometry, the corresponding variables for the original, undeformed mesh have a suffix `_mesh` (for example, `h_mesh`).

SPATIAL COORDINATE VARIABLES

- The spatial coordinate variables (independent variables) are available for all domain types.
- For a Cartesian geometry the default names for the spatial coordinates are x, y, z.
- For axisymmetric geometries the default names for the spatial coordinates are r, φ , and z.
- If a deformed mesh is used, x, y, z can be both the *spatial coordinates* (`x`, `y`, `z`) and the *material/reference coordinates* (`X`, `Y`, `Z`); see Mathematical Description of the Mesh Movement.
- If the model includes a deformed mesh, the variables x`TIME`, y`TIME`, z`TIME` represent the mesh velocity. To access these variables, replace x, y, and z with the names of the spatial coordinates in the model (`x`, `y`, and `z`).

PARAMETERIZATION VARIABLES

The surface-boundary parameterization variables can be useful for defining distributed loads and constraints such as a parabolic velocity profile. The available parameterization variables are:

The curve parameter `s` (or `s1`) in 2D. Use a line plot to visualize the range of the parameter, to see if the relationship between x and y (the spatial coordinates) and `s` is nonlinear, and to see if the curve parameterization is aligned with the direction of the corresponding boundary. In most cases it runs from 0 to 1 in the direction indicated by the arrows shown on the edges when in the boundary or edge selection mode and if you have selected the **Show edge direction arrows** check box in the **View** node's (xy) settings window. You can use `s` on boundaries in 2D when specifying boundary conditions.

The arc length parameter `s1` available on edges in 3D. It is approximately equivalent to the arc length of the edge. Use a line plot to visualize the values of `s1`.

The surface parameters `s1` and `s2` in 3D are available on boundaries (faces). They can be difficult to use because the relationship between x, y, and z (the spatial coordinates) and `s1` and `s2` is nonlinear. Often it is more convenient to use expressions with x, y, and z for specifying distributed boundary conditions. To see the values of `s1` and `s2`, plot them using a surface plot.

TANGENT AND NORMAL VARIABLES

The tangent and normal variables are components of the tangential and normal unit vectors.

Tangent Variables

In 2D, tx and ty define the curve tangent vector associated with the direction of the boundary.

In 3D, the tangent variables t1x, t1y, and t1z are defined on edges. The tangent variables t1x, t1y, t1z, t2x, t2y, and t2z are defined on surfaces according to

$$(t_{ix}, t_{iy}, t_{iz}) = k_i\left(\frac{\partial x(s_1, s_2)}{\partial s_i}, \frac{\partial y(s_1, s_2)}{\partial s_i}, \frac{\partial z(s_1, s_2)}{\partial s_i}\right), \quad i = 1, 2$$

These most often define two orthogonal vectors on a surface, but the orthogonality can be ruined by scaling geometry objects. The vectors are normalized; k_i is a normalizing parameter in the expression just given.

If a deformed mesh is used, the tangent variables are available both for the deformed configuration and for the undeformed configuration. In the first case, replace *x*, *y*, and *z* with the spatial coordinate names (`x`, `y`, and `z` by default). In the second case, replace *x*, *y*, and *z* with the material/reference coordinate names (`X`, `Y`, and `Z` by default).

Normal Variables

In 1D, nx is the outward unit normal pointing out from the domain.

In 2D, nx and ny define a normal vector pointing outward relative to the domains.

In 3D, nx, ny, and nz define a normal vector pointing outward relative to the domains.

Direction of the Normal Component on Interior Boundaries

To get control of the direction of the normal component on interior boundaries, the following variables are available:

In 1D:

- unx, the outward unit normal seen from the upper domain
- dnx, the outward unit normal seen from the lower domain

In 2D:

- un*x* and un*y* for the up direction
- dn*x* and dn*y* for the down direction

The upside is defined as the left side with respect to the direction of the boundary.

In 3D:

- un*x*, un*y*, and un*z* for the up direction
- dn*x*, dn*y*, and dn*z* for the down direction

To visualize any of these vector variables use arrow plots on surfaces or lines.

If a deformed mesh is used, the normal variables are available both for the deformed configuration and for the undeformed configuration. In the first case, replace *x*, *y*, and *z* with the spatial coordinate names (x, y, and z by default). In the second case, replace *x*, *y*, and *z* with the material/reference coordinate names (X, Y, and Z by default).

Normal Vector Variables Representing Element Surface Normals

A similar set of variables—n*x*mesh, un*x*mesh, and dn*x*mesh, where *x* is the name of a spatial coordinate—use the element shape function and are normal to the actual element surfaces rather than to the geometry surfaces.

CURVATURE VARIABLES

The curvature variables are defined on boundaries in 2D and 3D.

In 2D the curvature is denoted curv. Positive curvature is toward the normal (n*x*,n*y*).

In 3D there are two principal curvatures named curv1 and curv2, where curv1 is less than curv2 and seen as real numbers. These correspond to the minimal and maximal values for the curvature of a curve you get by intersecting the boundary with a plane in which the normal lies. Positive curvature is toward the normal (n*x*,n*y*,n*z*).

The components of the normalized tangential directions for the corresponding curvatures are called tcurv*x*, tcurv*y* in 2D and tcurv1*x*, tcurv1*y*, tcurv1*z*, tcurv2*x*, tcurv2*y*, and tcurv2*z* in 3D. The tangents (tcurv1*x*,tcurv1*y*,tcurv1*z*) and (tcurv2*x*,tcurv2*y*,tcurv2*z*) are orthogonal.

Note also that you can choose on which frame you evaluate all these variables by using the general instructions found below Table 5-9.

Shape Function Variables

A finite element defines a number of variables, typically a dependent variable and its derivatives. Such variables are called shape functions variables because they are computed directly from *shape functions* and the degrees of freedom.

When a physics interface is selected, you can enter names for the dependent variables—these names are used to construct the finite elements. The dependent variable name is the basis for additional variable names that the finite elements generate.

When entering the shape function variables, replace the letters highlighted below in *italic font* with the actual names for the dependent variables (solution components) and independent variables (spatial coordinates) for the Component.

For example, replace *u* with the names of the dependent variables in the individual Component, and replace *x*, *y*, and *z* with the first, second, and third spatial coordinate variable, respectively. x_i represents the *i*th spatial coordinate variable. If the Component contains a deformed mesh or the displacements control the spatial frame (in solid mechanics, for example), you can replace the symbols *x*, *y*, *z* with either the spatial coordinates (`x`, `y`, and `z` by default) or the material/reference coordinates (`X`, `Y`, and `Z` by default).

AN EXAMPLE OF LAGRANGE ELEMENT VARIABLES

For the Lagrange element, which is the element type used by most physics interfaces, Table 5-10 lists the available variable names, assuming you gave the name *u* as the argument to the shape function, and that the names *x*, *y*, and *z* are provided for the independent variables.

TABLE 5-10: LAGRANGE ELEMENT VARIABLE NAMES

ENTITY TYPE\ SPACE DIMENSION	1D	2D	3D
POINT		u	u
EDGE			u, uTx, uTy, uTz
BOUNDARY	u, uTx, ut, uTxt	u, uTx, uTy, ut, uTxt, uTyt	u, uTx, uTy, uTz, ut, uTxt, uTyt, uTzt
DOMAIN	u, ux, uxx, ut, uxt, uxxt, utt, uxtt, uxxtt,	u, ux, uy, uxx, uxy, uyx, uyy, ut, uxt, uyt, uxxt, uxyt, uyxt, uyyt, utt, uxtt, uytt, uxxtt, uxytt, uyxtt, uyytt	u, ux, uy, uz, uxx, uxy, uxz, uyx, uyy, uyz, uzx, uzy, uzz, ut, uxt, uyt, uzt, uxxt, uxyt, uxzt, uyxt, uyyt, uyzt, uzxt, uzyt, uzzt, utt, uxtt, uytt, uztt, uxxtt, uxytt, uxztt, uyxtt, uyytt, uyztt, uzxtt, uzytt, uzztt

For example, with a fluid-flow interface, you get the set of variables indicated in Table 5-10 for `u`, `v`, `w`, and `p`, respectively.

- The variables *ux*, *uy*, and *uz* are the components of the gradient ∇u, that is, the 1st-order spatial derivatives.
- The variables *uxx*, *uxy*, *uxz*, *uyx*, *uyy*, *uyz*, *uzx*, *uzy*, and *uzz* are the 2nd-order space derivative components. They are meaningful only if the degree of the polynomial shape function used in an element is high enough. For first-order elements all these variables evaluate to zero.
- For elements with 2nd-degree polynomial shape functions (2nd-order elements), the polynomial degree of the 2nd-order derivatives is zero; that is, the second derivatives are constant in each element.
- For element orders lower than two, the second derivatives evaluate to zero regardless of the values of the 2nd-order derivatives of the true solution.

If the model uses a deformed mesh, each finite element is associated with a certain frame (the spatial frame or the material frame). The frame determines the names of the variables generated by the finite element. For instance, if the spatial frame is used, the Lagrange element computes derivatives with respect to the spatial coordinates, `ux`, `uy`, and `uz`. If the material frame is used, the Lagrange element computes derivatives with respect to the material coordinates `uX`, `uY`, and `uZ`.

THE TIME DERIVATIVES OF THE DEPENDENT VARIABLES

The variable u`t` is the time derivative of the dependent variable u. You can also form mixed space-time derivatives as ux_i`t`, for example, `uxt`,

$$\frac{\partial^2 u}{\partial x \partial t}$$

! The `t` must be last in a mixed derivative. The second time derivatives can also be used, such as `utt` or `uxtt` (but not higher derivatives in time).

If the model contains a deformed mesh, there is, in addition to the usual time derivative `ut`, the *mesh time derivative* `uTIME`. This also holds for mixed space-time derivatives.

TANGENTIAL DERIVATIVE VARIABLES

On boundaries, edges, and points you also have access to *tangential derivative variables*. They have names such as u`T`x, u`T`y, and u`T`z. Using these variables, it is possible to create models with phenomena on boundaries, edges, or points as described with PDEs.

The tangential derivative variables represent the Cartesian components of the tangential projection of the gradient of shape function variables:

$$(\nabla u)_T = (I - \mathbf{n}\mathbf{n}^T) \cdot \nabla u$$

In this equation, $(\nabla u)_T$ is the tangential gradient, which consists of the tangential derivatives in each spatial direction, I is the unity tensor, $\mathbf{n}$ is the outward unit normal vector, and ∇u is the gradient of u.

LAGRANGE MULTIPLIER VARIABLES

If weak constraints are activated for boundary conditions that are constraints (Dirichlet boundary conditions), COMSOL adds variables for the Lagrange multipliers (one for each dependent variable) by adding `_lm` as a suffix to the dependent variable name. For example, for a dependent variable u, the corresponding Lagrange variable is `u_lm`. The Lagrange multipliers are available on boundaries, and you can also evaluate them on edges (in 3D) and points (in 2D and 3D).

VARIABLE INHERITANCE

On boundaries, edges, and points, gradients and second derivatives of the shape functions are available by *inheritance*; that is, the average of the values of the variables from the adjacent domains are computed. This process can progress for several levels.

For example, `ux` is the average on a boundary from the adjacent domains, then the average on an edge from the adjacent boundaries, and finally, the average at the points from the adjacent edges.

If possible, avoid using variable inheritance for gradients and second derivatives in a model. Instead, use the tangential derivative variables for equation-based modeling on boundaries.

For computations of integrals of reaction forces and fluxes, use the reacf operator.

For high accuracy reaction forces and fluxes in other circumstances, use weak constraints and Lagrange multipliers on boundaries instead of directly accessing the gradient through inheritance (see Computing Accurate Fluxes).

When you plot or evaluate—on a boundary, for example—the value of a variable that is discontinuous across that boundary (a thin resistive layer, for example), the value is the average of the value on the “up” and “down” sides of the boundary. You can use the `up` and `down` operators to get the value on either side of the boundary (see up and down).

Solver Variables

The following table lists global solver variables that are available during the solution process only. They can be used in solver settings in the Study branches but are not available for use in, for example, results evaluations and plots.

TABLE 5-11: SOLVER VARIABLES

VARIABLE	DESCRIPTION
`niterCMP`	This variable contains the iteration number for nonlinear iterations. It starts from one and increases with one for each fully coupled or segregated iteration. It is used by some physics to control damping mechanisms. Examples are pseudo-time stepping in fluid dynamics and the penalty factor in the augmented Lagrangian method for contact problems in structural mechanics.
`gmg_level`	This variable contains the geometric multigrid level. It is zero for the top level (the one solved for), one for the next coarser level, and so on. It is used by some physics to control artificial stabilization.
`timestep`	This variable contains the current time step used by the time-dependent solver. It is used by some physics to control artificial stabilization. You can use it, for example, to create a stop condition that stops the time stepping if the time step becomes smaller than some threshold value.

Entering Ranges and Vector-Valued Expressions

You can enter ranges and vector-valued expressions such as extra grid-line coordinates using the following formats:

- A space-separated or comma-separated list of values: `10, 15, 23, 29.7, 30`.
- A delimited space-separated list using curly braces; for example, as an argument to a function: `cos({0 pi/4 pi/2})`.
- Equally-spaced values using the `range` function as in `range(`*`start value`*`,`*`step_size`*`,`*`end value`*`)`. For example, `range(0,0.2,3)` creates the values 0, 0.2, 0.4,..., 2.6, 2.8, and 3.0. The step size is 1 if you provide only start and end values and skip the step value.

`start_value` can be either smaller or larger than *`end_value`*. In the latter case, the step size must be negative. For example, `range(0,-5,-100)` creates the values 0, −5, −10, ..., −95, −100, while `range(0,5,-100)` is an empty set of values.

Combine these formats in a single expression to create an array of values that contain an arbitrary number of segments with differently spaced values mixed with other freely specified values.

EXAMPLES USING THE RANGE FUNCTION

- `range(a,(b-a)/(n-1),b)` gives a list of *n* equally-spaced increasing values in the range [`a`, `b`] if `b >a` or decreasing values in the range [`b`, `a`] if `a >b`.
- `10^range(-3,3)` gives the exponentially increasing sequence 10^{-3}, 10^{-2}, …, 10^{3}.
- `1^range(1,10)` gives a sequence of length 10 where all elements equal 1. Multiplying the vector `1^range(1,n)` by a constant value `a` gives a vector of *n* elements all equal to `a`.
- `0^range(1,5)` gives the sequence `0 0 0 0 0`.

USING RANGES TO GENERATE ARRAYS

A convenient way to generate vectors of values is to use the **Range** dialog box, which you open by clicking the **Range** button ( ) next to most of the fields that accept vectors of values.

In that dialog box, use the **Entry method** list to select **Step** to enter a step size or **Number of values** to specify the number of values in the array. Specify the start value for an array of values in the **Start** field. Enter the step size in the **Step** field or the number of values in the **Number of values** field, depending on the setting in the **Entry method** list. Specify the end value for the array of values in the **Stop** field. By default, the spacing of the values is linear, but you can select a function to apply to all values. To do so, choose one of the available arithmetic and trigonometric functions from the **Function to apply to all values** list. For example, select **exp10** to create an array of exponentially increasing values. The list includes the following functions:

- The default value **None**, which means linear spacing using the `range` function directly with the values specified.
- The exponential functions **exp10** (base-10 exponential function) and **exp** (base-*e* exponential function), which create exponentially-spaced values using the specified range of values as powers of 10 and of the mathematical constant *e*, respectively.
- The trigonometric functions **cos** (cosine) and **sin** (sine), which create sinusoidally varying values.
- The square root function **sqrt**, which creates a vector with values that are the square roots of the values specified.

Click **Replace** to replace the contents in the field with the values specified in the **Range** dialog box.

Click **Add** to add the range of values to the end of the existing values in the associated field. That way you can create more complex ranges.

For ranges that contain integer values only, an **Integer Range** dialog box opens instead of the normal **Range** dialog box. The **Integer Range** dialog box only contains **Start**, **Step**, and **Stop** fields, all of which must contain integer values.

SUPPORT FOR RANGES AND VECTOR-VALUED EXPRESSIONS

The following modeling settings support ranges and vector-valued expressions:

- Extra grid lines in the **Axis** node's settings window.
- Interval coordinates when using the **Interval** node's settings window for 1D geometries.
- The **Copy**, **Move**, and **Rotate** transforms for geometry modeling.
- The times for output from the time-dependent solver and the list of parameter values in the settings windows for study step nodes for time-dependent and stationary solvers and for parametric sweeps.
- The contour levels, the streamline start-point coordinates, and the coordinates in arrow plots. Whenever you specify a number of coordinates in settings windows for plots, COMSOL uses scalar expansion—if one component is the same for all coordinates, enter a single number in the corresponding text field. For example, to get 101 linearly spaced coordinates from $y = 6$ to $y = 7$ along $x = 3$, enter it as the single scalar `3` for x and then `range(6,0.01,7)` for y. Thus, you need not enter 101 similar values for x.
- Element distribution in the meshing settings.

Summary of Built-In Variables With Reserved Names

This section is an overview of the built-in elements of the following categories as defined by the underlying COMSOL language:

- Constants
- Variables
- Functions

These language elements are built-in or user-defined. In addition there are *operators* that cannot be user-defined, and *expressions*, which are always user-defined.

ABOUT RESERVED NAMES

Built-in elements have reserved names, names that cannot be redefined by the user. If you try to use a reserved name for a user-defined variable, parameter, or function, the text where you enter the name turns orange and you get a tooltip error message if you select the text string. *Reserved function names are reserved only for function names, which means that such names can be used for variable and parameter names, and vice versa*. The following tables list the most commonly used built-in elements and hence those reserved names.

CONSTANTS AND PARAMETERS

There are three different types of constants: built-in *mathematical and numerical constants*, built-in *physical constants*, and *parameters*. Parameters are user-defined constants that can vary over parameter sweeps. Constants are scalar valued. The tables below list the built-in mathematical and numerical constants as well as built-in physical constants. Constants and parameters can have units.

BUILT-IN PHYSICAL CONSTANTS

NAME	DESCRIPTION
`g_const`	Acceleration of gravity
`N_A_const`	Avogadro constant
`k_B_const`	Boltzmann constant
`Z0_const`	Characteristic impedance of vacuum (impedance of free space)
`me_const`	Electron mass
`e_const`	Elementary charge
`F_const`	Faraday constant
`alpha_const`	Fine-structure constant
`G_const`	Gravitational constant
`V_m_const`	Molar volume of ideal gas (at 273.15 K and 1 atm)
`mn_const`	Neutron mass
`mu0_const`	Permeability of vacuum (magnetic constant)
`epsilon0_const`	Permittivity of vacuum (electric constant)
`h_const`	Planck's constant
`hbar_const`	Planck's constant over 2 pi
`mp_const`	Proton mass
`c_const`	Speed of light in vacuum
`sigma_const`	Stefan-Boltzmann constant
`R_const`	Universal gas constant
`b_const`	Wien displacement law constant

BUILT-IN MATHEMATICAL FUNCTIONS

These functions do not have units for their input or output arguments.

NAME	DESCRIPTION
`abs`	Absolute value
`acos`	Inverse cosine (in radians)
`acosh`	Inverse hyperbolic cosine
`acot`	Inverse cotangent (in radians)
`acoth`	Inverse hyperbolic cotangent
`acsc`	Inverse cosecant (in radians)
`acsch`	Inverse hyperbolic cosecant
`arg`	Phase angle (in radians)
`asec`	Inverse secant (in radians)
`asech`	Inverse hyperbolic secant
`asin`	Inverse sine (in radians)
`asinh`	Inverse hyperbolic sine
`atan`	Inverse tangent (in radians)
`atan2`	Four-quadrant inverse tangent (in radians)
`atanh`	Inverse hyperbolic tangent
`besselj`	Bessel function of the first kind
`bessely`	Bessel function of the second kind

NAME	DESCRIPTION
besseli	Modified Bessel function of the first kind
besselk	Modified Bessel function of the second kind
ceil	Nearest following integer
conj	Complex conjugate
cos	Cosine
cosh	Hyperbolic cosine
cot	Cotangent
coth	Hyperbolic cotangent
csc	Cosecant
csch	Hyperbolic cosecant
erf	Error function
exp	Exponential
floor	Nearest previous integer
gamma	Gamma function
imag	Imaginary part
log	Natural logarithm
log10	Base-10 logarithm
log2	Base-2 logarithm
max	Maximum of two arguments
min	Minimum of two arguments
mod	Modulo operator
psi	Psi function and its derivatives
range	Create a range of numbers
real	Real part
round	Round to closest integer
sec	Secant
sech	Hyperbolic secant
sign	Signum function
sin	Sine
sinh	Hyperbolic sine
sqrt	Square root
tan	Tangent
tanh	Hyperbolic tangent

The following tables summarize the built-in variables and functions that are generally available in all COMSOL models. Some are only available in certain geometries or in time-dependent models, for example. These variable names are reserved names and appear in orange in the settings windows for parameters and variables.

TABLE 5-12: BUILT-IN VARIABLES

NAME	DESCRIPTION	TYPE
t	Time	Scalar
freq	Frequency	Scalar
lambda	Eigenvalues	Scalar
phase	Phase angle	Scalar

TABLE 5-12: BUILT-IN VARIABLES

NAME	DESCRIPTION	TYPE
numberofdofs	Number of degrees of freedom	Scalar
x, y, z, r, X, Y, Z, R	Position	Field
s, s1, s2	Edge/surface parameters	Field
n, nx, ny, nz, nr	Edge/surface normals	Field
tx, ty, tz, tr	Edge tangents	Field
t1x, t1y, t1z, t2x, t2y, t2z	Surface tangents	Field
un, unx, uny, unz	Edge/surface upward normals	Field
dn, dnx, dny, dnz	Edge/surface downward normals	Field
eps, i, j, pi, inf, Inf, nan, NaN	Numerical constants	Scalar
h	Local mesh element size (length of the longest element edge)	Field
dom	The domain number, boundary number, edge number, or point number	Field
meshtype	Mesh type index for the mesh element; this is the number of edges in the element.	Field
meshelement	Mesh element number	Field
meshvol	Volume of the (linearized) mesh element	Field
dvol	Volume scale factor variable; this is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates.	Field
qual	A mesh quality measure between 0 (poor quality) and 1 (perfect quality)	Field
reldetjac	Determinant of the Jacobian matrix for the mapping from the straight mesh element to the possibly curved element used when solving	Field
reldetjacmin	The minimum value of the reldetjac variable in each element	Field
linearizedelem	One inside elements that have been linearized; zero otherwise	Field
niterCMP	Iteration number for nonlinear iterations	Scalar
gmg_level	Geometric multigrid level	Scalar
timestep	Current time step	Scalar

The suffixes x, y, z, and r in some of the variables are the default names for the spatial coordinates, which you can change if desired.

The following user-defined variables generate built-in variables such as space and time derivatives. See Shape Function Variables for information about those built-in variables.

TABLE 5-13: USER-DEFINED VARIABLES THAT GENERATE BUILT-IN VARIABLES

DEFAULT NAME	DESCRIPTION	TYPE
x, y, z	Spatial coordinates (Cartesian)	Field
r, z	Spatial coordinates (cylindrical)	Field
u, T, and so on	Dependent variables (solution)	Field

Mass Properties

Overview

You can add one or more **Mass Properties** nodes to create variables for and compute the following quantities for a model component:

- The volume of the geometry or part of the geometry.
- The mass of the geometry or part of the geometry
- The center of mass
- The moment of inertia
- The principal moment of inertia

When you compute the solution, COMSOL adds the following variables:

TABLE 5-14: VARIABLES CREATED BY A MASS PROPERTIES NODE

QUANTITY	TYPICAL VARIABLE NAMES
Volume	`mass1.volume`
Mass	`mass1.mass`
Center of mass	`mass1.CMX`, `mass1.CMY`, `mass1.CMZ`
Moment of inertia	`mass1.IXX`, `mass1.IXY`, `mass1.IXZ`, and so on
Moment of inertia, principal direction	`mass1.Ip1X`, `mass1.Ip2X`, `mass1.Ip3X`, and so on
Moment of inertia, principal values	`mass1.Ip1`, `mass1.Ip2`, and `mass1.Ip3X`

Use a **Global Evaluation** node, for example, to evaluate the resulting measurement quantities, which you can select from a **Definitions>Mass Properties** submenu after clicking the **Replace Expression** (▶) or the **Insert Expression** (✚) button.

Mass Properties

Right-click **Definitions** in a **Component** branch to add a **Mass Properties** node (⚖) to compute the variables for mass, volume, center of gravity, and moment of inertia. The settings window for a **Mass Properties** node includes the following sections:

SOURCE SELECTION

The source selection defines the source for the mass property variables—the part of the geometry over which the program computes the measurement variables and to which the density contribution is limited. you can add other density contributions by right-clicking the **Mass Properties** node and choose **Mass Contributions** (⚖); see Mass Contributions.

From the **Geometric entity level** list, select **Domain** (the default), **Boundary**, **Edge** (3D only), or **Point**. Select **Manual** or **All domains** (the default), **All boundaries**, **All edges**, or **All points** from the **Selection** list. If **Manual** is selected, select geometric entities in the **Graphics** window.

DENSITY

In this section you define the source of the density values used for computing some of the mass properties. In the **Density expression** field, type a value or expression for the density (SI unit: kg/m^3). For example, `material.rho`, which is a variable for the density from the materials in the model.

From the **Integration frame** list, choose the frame to use for the integration. The default, **Material (X, Y, Z)** in 3D, uses the material frame.

VARIABLES

In the **Variables** section you select which variables to create and on which frame they are defined.

From the **Frame** list, choose the frame to use when computing the mass properties. The default, **Material (X, Y, Z)** in 3D, uses the material frame.

Select the following check boxes to create and compute the corresponding mass property (by default, all of them are selected):

- **Create volume variable** to create a variable for the volume of the selected geometric entities.
- **Create mass variable** to create a variable for the mass of the selected geometric entities.
- **Create center of mass variables** to create variables for the center of mass of the selected geometric entities.
- **Create moment of inertia variables** to create variables for the moment of inertia of the selected geometric entities.
- **Create principal moment of inertia variables** to create a variables for the principal moment of inertia of the selected geometric entities.

INTEGRATION SETTINGS

In this section you can specify the integration order of the integration used to compute the output variables. The default in the **Integration order** field is 4 (which typically is twice the order of the shape order function for the physics).

Mass Contributions

Right-click a **Mass Properties** node ( ) to add a **Mass Contributions** ( ) node, which you can use to add contributions to the mass used for computing the mass properties in the parent **Mass Properties** node. The contributions can come from a connected boundary, for example. The settings window for a **Mass Contribution** node includes the following sections:

SOURCE SELECTION

The source selection defines the source for the mass contribution to the mass properties—the part of the geometry where you want to add mass contributions.

From the **Geometric entity level** list, select **Domain** (the default), **Boundary**, **Edge** (3D only), or **Point**. Select **Manual** or **All domains** (the default), **All boundaries**, **All edges**, or **All points** from the **Selection** list. If **Manual** is selected, select geometric entities in the **Graphics** window.

DENSITY

In this section you define the source of the density values used for computing some of the mass properties. In the **Density expression** field, type a value or expression for the density (SI unit: kg/m^3).

From the **Integration frame** list, choose the frame to use for the integration. The default, **Material (X, Y, Z)** in 3D, uses the material frame.

Functions

User-defined functions can be added globally or locally:

- To add global functions, on the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select an option from the **Functions** menu, or right-click the **Global Definitions** (≡) node.
- You can add local functions (for any **Component** branch), from two toolbars, either the **Definitions** toolbar in the **Functions** group, or on the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users), where you choose it from the **Functions>Local** submenu. You can also right-click the **Definitions** (≡) node and choose an option from the **Functions** submenu.

About User-Defined Functions

There are three broad categories of user-defined functions—**Analytic**, **Interpolation**, and **Piecewise**—and a number of templates for common function types, such as step and ramp functions. You can also create external function interfaces to include functions written in C and MATLAB. Functions can be global or local in a model component, although *external functions* and *MATLAB functions* can only be defined globally.

NAMING FUNCTIONS

Function names for built-in mathematical functions such as `abs`, `cos`, and `test` are reserved function names, and naming a user-defined function using one of the reserved function names is not recommended because it can cause unexpected results. If the name that you type in the **Function name** text field is a reserved function name, the text color changes to orange, as a warning. If you move the cursor to a function name in orange, the tooltip **is a reserved name** is displayed.

PLOTTING FUNCTIONS

Click the **Plot** button ( ) in the upper-right corner of the settings window to plot any user-defined function of 1–3 variables directly in the Graphics window. This plot is temporary and disappears when you move to another Model Builder node making use of the Graphics window.

Click the **Create Plot** button ( ) in the upper-right corner of the settings window to create a persistent plot of the function under Results, including a Function data set, a plot group and a plot feature. The added nodes can be used for plotting multiple functions on top of each other or for direct comparison to model results, for example.

For analytic functions, first define a range for the arguments in the **Plot Parameters** section. Note that the range is expressed in the current unit system's base unit corresponding to the set function argument unit. For example, if the current unit system is SI and the function argument unit is `cm`, the plot range is expressed in `m`.

UNITS FOR FUNCTION INPUTS AND OUTPUTS

By default, functions expect dimensionless inputs and provide dimensionless outputs. When such a function is called with arguments having well-defined units (except when explicitly dimensionless), a unit warning will be displayed. If the function is used anyway, it will be called for the numerical value of the argument expressed in the current unit system for the context where function is called. The return value will from the unit handling perspective behave like a number, adapting its unit to the context where it is used.

In the **Units** section in the settings window for the Analytic, Interpolation, and Piecewise function nodes, you can define units for the function's inputs and output. In the **Arguments** field, type a single unit for all inputs, or specify space-separated or comma-separated units for each input (type `m, s`, for example, for two input arguments with the units meter and second, respectively). In the **Function** field, type the unit for the function's output.

Common Settings for the Function Nodes

UNITS

By default, functions expect dimensionless inputs and provide dimensionless outputs. In the **Units** section in the settings window for the Analytic, Interpolation, and Piecewise function nodes, you can define units for the function's inputs and output. In the **Arguments** field, type a single unit for all inputs, or specify space-separated or comma-separated units for each input (type `m, s`, for example, for two input arguments with the units meter and second, respectively). In the **Function** field, type the unit for the function's output.

DERIVATIVES

For External functions and MATLAB functions, enter expressions for the partial derivatives of each function with respect to their input arguments. The derivative information is needed by the automatic Jacobian computation. In the **Derivatives** table, each row contains a partial derivative of a function with respect to one of its arguments:

- The entries in the **Function name** column must occur in the **Functions** table, and the entries in the **Argument** column must occur among the arguments listed for that function in the **Function** table.
- The **Partial derivative** column contains expressions for the partial derivatives. Partial derivatives that are not specified in the table default to 0.

PLOT PARAMETERS

Use this table to set the range for arguments in preview plots. For each argument, enter a **Lower limit**, and an **Upper limit** in the **Plot Parameters** table. In Analytic function plot settings, the argument column is updated automatically and must always match the specified argument list. For External and MATLAB functions, the argument column is absent. Instead, the limits specified in the table are applied to in order from top to bottom to the arguments of the first function with a matching number of arguments. It is also this first matching function which is plotted when you click the **Plot** button () or the **Create Plot** button ().

SMOOTHING

Smoothing makes a function more well-behaved and suitable for modeling. It replaces jumps with smooth transitions that eliminate discontinuities and can represent the inertia present in real applications. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. The smoothed functions have continuous first and, optionally, second derivatives.

For the Ramp, Step, Triangle, and Rectangle functions, enter a value in the **Size of transition zone** field to control the amount of smoothing. Set the **Number of Continuous Derivatives** to **1** or **2**. The default is to make derivatives continuous up to second order. For the Ramp function, additionally choose whether to **Smooth at start** and/or **Smooth at cutoff**.

Note that smoothing affects for which values of the arguments the function is different from zero. In particular, when applying smoothing to a Ramp or Step function, it will start to rise before the position specified in the **Location** field. This may interfere with initialization assuming that the function is zero at this point.

The Waveform function also supports smoothing for some of the waveform types.

Analytic

An **Analytic** function (f(x)) is defined by a symbolic expression. Analytic functions have the ability to bind arguments during function calls. In other words, they do not require the actual argument names in an expression when writing the function. For example, you can define a function $f(x) = x^2$ with the input argument `x` and the expression `x^2` and the call it as `f(T)`, where `T` is the temperature in a heat transfer model. The default **Function name** is `an1`.

To add a global or local (for any Component) Analytic function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Analytic** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** (≡) node or the **Definitions** (≡) node and choose it from the **Functions** submenu.

- See Naming Functions for information about allowed function names.
- See Units for information about the **Unit** section.
- See Plot Parameters for information about plot range settings.

DEFINITION

In the **Expression** field, enter the mathematical expression that defines the function, such as `sin(x)*cos(y)+g_const` or `a+b*cos(c)`. Enter **Arguments** to the analytic function as comma-separated entries (`x`, `y` and `a`, `b`, `c` for the functions above). In addition to the arguments that are defined, analytic functions also recognize global parameters and physical constants.

From the **Derivatives** list, **Automatic** is selected by default and computes the derivatives symbolically. COMSOL uses the derivatives of a function if a variable that depends on the solution is used in a function argument. Select **Manual** to specify the function derivatives with respect to its arguments in a table. If **Manual** is selected, enter the derivatives with respect to the function's arguments. For undefined derivatives, COMSOL uses 0 as the value of the derivative. In the second example above, enter `a`, `b`, and `c` in the top three rows of the **Argument** column, and `1`, `cos(c)`, and `-b*sin(c)`in the associated text fields in the **Partial derivative** column.

PERIODIC EXTENSION

Select the **Make periodic** check box to make the function periodic and extend its definition within an interval to the whole real axis. Then define the interval by entering values in the **Lower limit** (default is 0) and **Upper limit** (default is 1) fields.

ADVANCED

Select the **May produce complex output for real arguments** check box if the defined function works similarly to `sqrt`; that is, if it sometimes returns complex values for a real-valued input.

- If you have the AC/DC Module, see Geoelectrics: model library path **ACDC_Module/Other_Industrial_Applications/geoelectrics**.
- If you have the RF Module, see Second Harmonic Generation of a Gaussian Beam: model library path **RF_Module/Tutorial_Models/second_harmonic_generation**.

Elevation

The **Elevation** function ( ) makes it possible to import geospatial elevation data from digital elevation models (on the DEM file format using the USGS standard from the United States Geological Survey) and map the elevation data to a function of *x* and *y*. A DEM file contains elevation data for a portion of the Earth's surface. The resulting function behaves essentially like a grid-based interpolation function. The default **Function name** is `elev1`.

To add a global or local (for any Component) Elevation function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Elevation (DEM)** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** (☰) node or the **Definitions** (☰) node and choose it from the **Functions** submenu.

FILE

Enter the path and name of the elevation file in the **Filename** text field, or click **Browse** to select a DEM file with elevation data in the **Elevation Data** dialog box. When a DEM file is open, the **File** section displays the coordinates for the southeast corner.

Click **Import** to import the elevation data in the specified DEM file into the model; otherwise, COMSOL references the elevation data on your file system. When the elevation data is imported, the **File** section (under **Data imported into model**) contains information about the filename and the location for the data. Click **Export** to save the elevation data to a file and reference from that file instead of including it in the model. Click the **Discard** button to delete the imported data from the model.

INTERPOLATION AND EXTRAPOLATION

For interpolation in the elevation data, select an **Interpolation** method from the list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the elevation data, select an **Extrapolation** method from the list—**Constant** (the default), **Linear**, **Nearest function** (which evaluates the function from the closes grid point at the actual point where a value is requested), or **Specific value**. For a **Specific value**, enter a value to **Replace missing data with** field (SI unit: m). The default is 0 m.

On the settings window toolbar, click the **Create Surface** button ( ) to add a **Parametric Surface** node to represent the elevation function as a parametric surface in the **Geometry** branch for 3D models.

External

The **External** function is only available with the **Global Definitions** node. See also `model.func()` in the *COMSOL API Reference Manual* for details about the interface to external functions, including an example and information about compiling the function on the supported platforms.

An **External** function ( ) interfaces to other external functions written in the C language (using a native C function or, through a wrapper function, interfacing with source code written in, for example, Fortran). Then use those functions as any other functions defined in COMSOL. For example, use it for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file).

To add a global External function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **External** from the **Functions>Global** menu, or
- Right-click the **Global Definitions** (≡) node and choose it from the **Functions** submenu.

Go to Common Settings for the Function Nodes for information about the **Derivatives** and **Plot Parameters** sections.

FUNCTIONS

Enter a **Library** path and name (the complete network path), or click **Browse** to locate a library to import. For each row in the table, enter a **Function name** (`myfun`, for example) and a space-separated or comma-separated list of the names of its input **Arguments** (`x y`, for example).

ADVANCED

Enter a value in the **Initialization data** field. The value is sent to the library when it is loaded. Select the **Thread safe** check box to declare that the function is a thread-safe pure function (that is, a function that always returns the same results using the same input argument values and that do not have any side effects or output). Select this check box to then improve performance.

Gaussian Pulse

The **Gaussian Pulse** function ( ) is the common bell-shaped curve (Gaussian function). It has a shape that is similar to a Gaussian (normal) distribution. The default **Function name** is `gp1`.

To add a global or local (for any Component) Gaussian Pulse function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Gaussian Pulse** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** (≡) node or the **Definitions** (≡) node and choose it from the **Functions** submenu.

The Gaussian pulse has the same characteristics as the normal distribution: it is a pulse with a shape that is similar to a normal or Gaussian distribution as a function:

$$y(x)= \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-x_0)^2}{2\sigma^2}}$$

In the equation above, x is the input variable, x_0 is the location (mean), and σ is the standard deviation. This function is a function of one variable (the time t, for example).

PARAMETERS

Enter a **Location** value for the Gaussian pulse mean x_0 (the default location is 0). Enter a **Standard deviation** σ of the normal distribution. The default is 1.

- If you have the RF Module, see Transient Modeling of a Coaxial Cable: model library path **RF_Module/Verification_Models/coaxial_cable_transient**.
- If you have the Chemical Reaction Engineering Module, see Surface Reactions in a Biosensor: model library path **Chemical_Reaction_Engineering_Module/Surface_Reactions_and_Deposition_Processes/reacting_pillars**.

Image

The **Image** function ( ) makes it possible to import an image (in BMP, JPEG, PNG, or GIF format) and map the image's RGB (red, green, blue) data to a scalar (single channel) function output value. By default the function's output uses the mapping (R+G+B)/3.

The default **Function name** is `im1`. An image is defined on a two-dimensional domain, and you typically call the image function using spatial coordinates: `im1(x,y)`.

To add a global or local (for any Component) Image function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Image** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

- See Naming Functions for information about the **Function Name** section.
- See Common Settings for the Function Nodes for information about the **Units** section.

FILE

Enter the path and name of the image file in the **Filename** text field, or click **Browse** to select an image file in the **Image** dialog box. Click **Import** to import the image in the specified image file into the model; otherwise COMSOL references the image on your file system. When you have imported the image, the **File** section, under **Data imported into model**, contains information about the image's filename and size. Click **Export** to save the image to a file and reference it from there instead of keeping it in the model. Click the **Discard** button to delete the imported image data from the model.

COORDINATES

Define the 2D coordinates and if required, flip the image. Select the **In place** check box to use the pixels in the image as the coordinates. Click to clear the check box to define the image coordinates explicitly using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields. Select the **Flip horizontally** check box to flip the image horizontally from left to right and vice versa. Select the **Flip vertically** check box to flip the image vertically from up to down and vice versa.

COLOR SCALING

From the **Scaling** list, select **Automatic** (the default) to use the default scaling, which outputs the mean of the RGB values for each pixel in the image. Select **Manual** to specify a custom expression for the scalar image function output value in the **Expression** field. The default is `(r+g+b)/3`, which is the automatic scaling, giving a scalar value that is the mean of the RGB values in each pixel.

For interpolation in the image, select an **Interpolation** method from the list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the image, select an **Extrapolation** method from the list—**Constant** (the default), **Linear**, **Nearest function** (which evaluates the function from the closes grid point at the actual point where a value is requested), or **Specific value**. If **Specific value** is selected, enter a value in the **Value outside range** field (default value: 0).

CLIPPING

Apply clipping to create a box-shaped region inside of the original image where the image is rendered. From the **Clipping** list, select **None** (the default) for no clipping, or **Manual** to define the box-shaped region using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields (unit: px). The default **x maximum** and **y maximum** values are both 1000 px; the default minimum values are 0.

Interpolation

An **Interpolation** function ( ) is defined by a table or file containing the values of the function in discrete points. The interpolation data can be structured (defined on a grid) or unstructured (defined on a generic point cloud).

To add a global or local (for any Component) Interpolation function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Interpolation** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

DEFINITION

Select a **Data source**—**Table** (the default) or **File** to define the interpolation function by entering values in a table or by importing interpolation data from a file, respectively.

- If **Table** is selected, enter a **Function name** and enter coordinates t and function values $f(t)$ into the table cells. A function of one variable can be defined in this way. For functions of two or more variables, such as space-dependent data in 2D and 3D, use a file with the function data. The default **Function name** is `int1`.
- *Optional:* Save the parameters to a text file to reuse in other models. Click the **Save to File** button ( ) and enter a **File name**, including the extension .txt. Click to **Save** the text file. The information is saved in space-separated columns in the same order as displayed on screen.
- *Optional*: Import or Load data from a spreadsheet program or other format by saving the file with a .txt extension. Data must be separated by tabs or colons. Click the **Load from File** button ( ) and navigate to the text (.txt) file to load and click **Open**. If the license includes LiveLink™ *for* Excel® you can also load interpolation data from a Microsoft Excel Workbook spreadsheet.The data loads to the table. If there is other data already in the table, it adds it after the last row. Move or edit as required.
- *Optional* (for functions defined by a table only): If you want to define the inverse function f^{-1} for the interpolation function f, select the **Define inverse function** check box and enter a function name for the inverse function in the **Inverse function name** field (the default is the name of the interpolation function with a suffix `_inv`). If you want to plot the inverse function instead of the interpolation function itself, first select the **Plot the**

inverse function check box in the **Plot Parameters** sections, which is only available when you have chosen to define the inverse function.

The inverse function only exists if the function is strictly monotonic.

If **File** is selected to import interpolation data from a file, select a **Data format**—**Spreadsheet**, **Grid**, or **Sectionwise**. If the license includes LiveLink™ *for* Excel®, you can also import interpolation data from a Microsoft® Excel Workbook spreadsheet. COMSOL then uses the spreadsheet format and the **Data format** list is not available.

- Enter a **Filename** (the complete network path) or **Browse** to locate a file to import.
- From the **Data format** list select **Spreadsheet**, **Grid**, or **Sectionwise**. The spreadsheet format is the default format, and that format is the easiest to use for functions defined on an unstructured grid or for general tabulated function values with one or more arguments.

If **Spreadsheet** is selected, enter a **Number of arguments** (1–3). For all file types, enter information about the functions into the table. Add a **Function** name and its **Position in file**. The first function in the file has position 1, the following has position 2, and so on. For spreadsheet data, the first columns contain the arguments (typically spatial coordinates); the following columns can contain one or more functions, and the positions entered are the relative position for each function's data column.

When the data format is specified, enter the path and name of the interpolation data file in the **Filename** text field, or click **Browse** to select a text or data file with interpolation data in the **Interpolation Data** dialog box. Then click **Import** to import the interpolation data into the model; otherwise COMSOL references the interpolation data on your file system. When you have imported the interpolation data, the **Parameters** section, under **Data imported into model**, contains information about the filename, data type, and dimension for the data. Click **Export** to save the interpolation data to a file and reference from there instead of including it in the model. Click the **Discard** button to delete the imported interpolation data from the model.

For the common case where the data source contains function values that are functions of the spatial coordinates, select the **Use space coordinates as arguments** check box. Then select the frame to which the spatial coordinates are attached from the **Frame** list (the default is **Spatial** for the spatial frame). Then the function can be called without arguments when used in the model; the spatial coordinates are added as function arguments automatically. The **Use space coordinates as arguments** check box is available for **Interpolation** nodes in a **Component** branch when the **Data source** is **File** or when using a **Table** in 1D models.

The Interpolation functions support 1, 2, or 3 arguments. You cannot define functions with more than three (3) arguments because the algorithm creates a mesh for the point cloud, which is not possible in four dimensions or higher.

An Example of Importing a File Data Source into a Parameter Table

The file named `temp.txt` contains temperature measurements in nine points in the plane:

```
10 3 310
20 3 309
30 3 314
10 6 302
20 6 307
30 6 311
10 9 307
20 9 308
```

```
20 9 314
```

The data columns contain *x*-coordinates, *y*-coordinates, and temperature values, respectively. To use this file as an interpolation function called `tempfun`, perform the following steps.

1. Select **File** from the **Data source** list.
2. Enter a **Filename** (the complete network path) or **Browse** to locate a file to import.
3. From the **Data format** list select **Spreadsheet**.
4. Enter a **Number of arguments**. In this example, enter `2`.
5. Enter the **Function** name `tempfun`.
6. Enter its **Position in file** as `1`. The first function in the file has position 1, the following has position 2, and so on. The position in file for a function is the column after the spatial coordinates (or other function arguments) where it is defined. In this example with two arguments (spatial coordinates), the third column is Position 1 in the file.
7. If desired, adjust the interpolation and extrapolation settings in the **Interpolation and Extrapolation** section (see below).

Use the function `tempfun` with `x` and `y` as input arguments in a 2D model to get the interpolated value for the temperature at any position. If the **Use space coordinates as arguments** check box is selected, use `tempfun` without adding the input arguments.

Examples of Spreadsheet, Sectionwise, and Grid File Formats

Spreadsheet File A *spreadsheet* file contains unstructured mesh and function data for space-dependent functions or general input variables and function values for functions of one or more variables.

```
    %Header (optional)
    Columns containing x, y (optional), and z (optional), or any other arguments, followed
by function data columns.
```

Include function names in the header. In that case, the input columns must be labeled with `x`, `y`, and `z`, respectively, depending on input dimension. For example, a file with the following content creates two 2D functions named `myfun1` and `myfun2`:

```
% x y myfun1 myfun2
0 0 0.12 0.34
0 1 0.52 1.50
1 0 0.67 0.91
```

If the file does not include any header to indicate the function dimension, the software assumes that it is identical to the largest geometry dimension present in the model. A file with four columns, for example, is interpreted as one 3D function in a 3D Component model, two 2D functions in a 2D Component model, and three 1D functions in a 1D Component model.

Sectionwise File A *sectionwise* file has coordinates and function values.

```
%Coordinates
One to three columns containing x, y (optional), and
z (optional)
%Elements
Triangulation where each row contains the row indices of the
points in the Coordinates section that make up one element
(triangular in 2D, tetrahedral in 3D)
%Data (funname)
Column of data values for each point
```

It is possible to include more than one function in the file as long as a `%Data` header separates them one from the other.

Grid File A *grid* file

```
%Grid
x grid points separated by spaces
y grid points separated by spaces (optional)
z grid points separated by spaces (optional)
%Data
Data values separated by spaces
```

Each row contains values for different x grid points for fixed values of y and z. The rows first increase the y grid value and then the z grid value. The grid points can also represent another independent variable that the data values depend on. For example, the "grid points" can be temperature values and the data values the thermal conductivity at these temperatures.

It is possible to include more than one function in the file as long as a `%Data` header separates them one from the other.

!	It is important to use a comment line starting with `%` to separate the grid points or other interpolation points and the data values that are associated with these coordinates or interpolation points.

INTERPOLATION AND EXTRAPOLATION

The interpolation and extrapolation settings control how the program evaluates the function between the discrete points where it is defined by the table or file, and the behavior of the function outside the domain where it is defined by the table or file.

Select an **Interpolation** method:

- For functions of one variable select **Nearest neighbor**, **Linear** (the default interpolation method), **Piecewise cubic**, or **Cubic spline**.
  - **Nearest neighbor** interpolation selects the value of the nearest point where the function is defined.
  - **Linear interpolation** uses a linear polynomial to interpolate the function between the points where it is defined.
  - **Piecewise cubic** interpolation uses a piecewise-cubic Hermite polynomial with continuous first derivatives. It preserves the shape of the data and respects monotonicity.
  - The **Cubic spline** method also performs interpolation with a piecewise cubic polynomial. Here, even second derivatives are continuous; however, the method does not necessarily respect monotonicity.
- For functions of more than one variable, select **Nearest neighbor** or **Linear**. The other options are not supported.

Select an **Extrapolation** method to specify how to treat arguments outside the grid or mesh of points.

- **Constant**. Uses the value from the closest point inside the grid (for structured interpolation) or the value from the closest mesh element (for unstructured interpolation). The function evaluates the polynomial from the closest grid point at the actual point where a value is requested. This is the default extrapolation method.
- **Linear**. The function is linear outside the grid with a continuous function value and continuous first derivative at the boundary of the grid. **Piecewise cubic** or **Cubic spline** must be selected from the **Interpolation** list.

- **Nearest function**. Evaluates the polynomial from the closest grid point at the actual point where a value is requested.
- **Specific value**. Uses a single value, such as zero or NaN (Not-a-Number), everywhere outside the grid or mesh. Enter the value in the **Values outside range** field.

Unstructured extrapolation supports a constant or a specific value only.

PLOT PARAMETERS

If you have selected to define the inverse function, this section becomes available. Select the **Plot the inverse function** check box to plot the inverse function instead of the interpolation function itself when you click the **Plot** button ( ) to generate a preview plot.

A plot group for an interpolation function contains three connected line graphs: a line graph of the piecewise function and two line graphs using dashed red lines, Left Extrapolation and Right Extrapolation, which show how the selected extrapolation extends the interpolation function on both sides. In addition, a point graph shows the interpolation points (interpolation nodes).

- See Common Settings for the Function Nodes for information about the **Units** section.

If you have the:

- Acoustics Module, see Muffler with Perforates: model library path **Acoustics_Module/Industrial_Models/perforated_muffler**.
- CFD Module, see Transonic Flow in a Sajben Diffuser: model library path **CFD_Module/High_Mach_Number_Flow/sajben_diffuser**.
- Corrosion Module, see Cathodic Protection of Steel in Reinforced Concrete: model library path **Corrosion_Module/Cathodic_Protection/cathodic_protection_in_concrete**.
- Heat Transfer Module, see Temperature Field in a Cooling Flange: model library path **Heat_Transfer_Module/Thermal_Processing/cooling_flange**.
- Molecular Flow Module, see Knudsen's Minimum: model library path **Molecular_Flow_Module/Benchmark_Models/knudsen_minimum**.
- Pipe Flow Module, see Geothermal Heating from a Pond Loop: model library path **Pipe_Flow_Module/Heat_Transfer/geothermal_heating**.
- Structural Mechanics Module, see Nonlinear Magnetostrictive Transducer: model library path **Structural_Mechanics_Module/Magnetostrictive_Devices/nonlinear_magnetostriction**.

MATLAB

> A LiveLink™ *for* MATLAB® license is required. Also confirm that MATLAB is installed on the same computer as COMSOL Multiphysics.

Use **MATLAB** functions node ( ) from COMSOL to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL. MATLAB functions are evaluated in MATLAB. You select a **MATLAB** function node from the **Functions** submenu.

FUNCTIONS

Under **Functions**, add the names of the MATLAB functions in the **Function name** column, and for each function in the table, enter a space-separated or comma-separated list of the names of its input **Arguments** (`x y`, for example). Click the **Clear Functions** button to force a reload of the functions if they have been edited. Alternatively, you can select the **Clear functions automatically before solving** check box.

> - Go to Common Settings for the Function Nodes for information about the **Derivatives** and **Plot Parameters** sections.
> - The *LiveLink™ for MATLAB® User's Guide*

Piecewise

A **Piecewise** function ( ) is created by splicing together several functions, each defined on one interval. Define the argument, extrapolation and smoothing methods, and the functions and their intervals. The piecewise function is a function of one variable with different definitions on different intervals, which must not overlap or have any holes between them. The default **Function name** is `pw1`.

To add a global or local (for any Component) Piecewise function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Piecewise** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

DEFINITION

Enter a name for the argument to the piecewise function in the **Argument** field.

Select an **Extrapolation** method to control what happens when the function is evaluated in a point that does not belong to any interval— **Constant** (the default), **None**, **Nearest function**, **Specific value**, or **Periodic**.

- **Constant**. Uses the function value from the endpoint of the closest interval. Uses the value from the start point of the first and the end point of the last interval on the corresponding sides.
- **None**. Evaluation fails outside of the intervals where it is defined. Trying to evaluate the function generates an error and evaluates to NaN (Not-a-Number).
- **Nearest function**. Evaluate the function from the closest interval. The function at the first or last interval (depending of the side) is used to evaluate at the actual point where a value is requested.
- **Specific value**. Also enter a value in the **Value outside range** field. If selecting the specific-number method, assign a single value (usually zero or NaN) to all points outside the intervals.

- **Periodic**. The function becomes periodic by repeating its values in the interval where it is defined in regular intervals of the same size.

Select a **Smoothing**: **No smoothing** (the default), **Continuous function** (to make the function continuous but not its derivatives), **Continuous first derivative**, or **Continuous second derivative**. For any selection (except **No smoothing**), enter a value in the **Relative size of transition zone** field (dimensionless). The default is 0.1. Relative size means relative in relation to the size of the intervals on both sides of the border.

The functions in contiguous intervals need not evaluate to the same value where the intervals meet. If the values differ the function has a discontinuity, and it is recommended that you apply smoothing to make the piecewise function more well-behaved numerically. Functions from neighboring intervals are then blended close to where the intervals meet.

For each cell in the **Intervals** table, enter **Start** and **End** interval limits. The intervals must not overlap, and there cannot be any holes between intervals. Enter an expression defining the **Function**.

The intervals must be contiguous.

A plot group for a piecewise function contains three connected line graphs: a line graph of the piecewise function and two line graphs using dashed red lines, Left Extrapolation and Right Extrapolation, that show how the selected extrapolation extends the piecewise function on both sides.

See Common Settings for the Function Nodes for information about the **Units** section.

If you have the Heat Transfer Module, see Radiative Heat Transfer in a Utility Boiler: model library path **Heat_Transfer_Module/Thermal_Radiation/boiler**.

Ramp

A **Ramp** function ( ) is a linear increase with a user-defined slope that begins at some specified time. The ramp function is a function of one variable (the time t, for example). The default **Function name** is `rm1`.

To add a global or local (for any Component) Ramp function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Ramp** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

- See Naming Functions for information about the **Function Name** section.
- See Common Settings for the Function Nodes for information about the **Smoothing** section.

PARAMETERS

Enter a **Location** value s_0 for the start of the ramp. The function evaluates to 0 for values less than its start location and increases linearly for values greater than the location.

If the **Smooth at start** check box is selected under **Smoothing**, then the **Ramp** function can become nonzero at some time before the location value. A model might not solve due to the possible inconsistencies between the ramp excitation and the initial condition (for example, the initial condition is zero, but the ramp function returns a small nonzero value at the starting time for the simulation).

Enter a **Slope** k (dimensionless) of the ramp. The default is 1.

To ensure that the value never exceeds a certain point, select the **Cutoff** check box and enter a value. The default is 1. For an input variable s, a start location s_0, and a slope k, the ramp function's value is 0 for $s<s_0$ and $k(s-s_0)$ for $s \geq s_0$.

Random

A **Random** function ( ) generates white noise with uniform or normal distribution and has one or more arguments to simulate white noise. Its distribution is either uniform or normal. The default **Function name** is `rn1`.

To generate a random function of the spatial coordinates x, y, and z, for example, use this function with three input variables—it returns the same value each time it is called for a given set of input arguments. To do a Monte Carlo simulation, add one or more additional input arguments that vary during a parametric sweep, for example.

To add a global or local (for any Component) Random function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Random** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

See Naming Functions for information about the **Function Name** section.

PARAMETERS

Enter a **Number of arguments** to the random function (the default is 1). Select a **Distribution** method—**Uniform** (the default) or **Normal**. If **Uniform** is selected, enter a **Mean** and **Range**. The default mean and range is 0 and 1, respectively. The range is the difference between the largest and smallest values that the function can return. If **Normal** is selected, enter a **Mean** and **Standard deviation**. The defaults are 0 and 1, respectively.

Rectangle

A **Rectangle** function ( ) is `1` in an interval and `0` everywhere else. This function (also called top hat or boxcar) is useful for band-pass filtering; you can use it to select values within an interval. It can also simulate a signal that is turned on during an interval or a load that is active on a part of a boundary, for example. The rectangle function is a function of one variable (the time t, for example). The default **Function name** is `rect1`.

To add a global or local (for any Component) Rectangle function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Rectangle** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

LIMITS

Enter a **Lower limit** (the default is −0.5) and **Upper limit** (the default is 0.5) to specify the interval for the rectangle function. For example, if the input argument is time, enter a start and end time. This function evaluates to 1 for values within the interval from the lower limit to the upper limit. Outside the interval it evaluates to 0.

- See Naming Functions for information about the **Function Name** section.
- See Common Settings for the Function Nodes for information about the **Smoothing** section.

If you have the Batteries & Fuel Cells Module, see Soluble Lead-Acid Redox Flow Battery: model library path **Batteries_and_Fuel_Cells_Module/Batteries/pb_flow_battery**.

Step

A **Step** function ( ) is a sharp transition from `0` to some other value (amplitude) at some location (a certain time, for example). Create a single step function with a certain amplitude from an initial level to a final level at a start location. The step function is a function of one variable (the time t, for example). The default **Function name** is `step1`.

To add a global or local (for any Component) Step function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Step** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

PARAMETERS

Enter a **Location** (s_0) of the step. The value of the step function is the initial level for input values that are smaller than the location of the step. In the **From** field, enter a start level (L_{start}). In the **To** field, enter a final level (L_{end}). For an input variable s, a start location s_0, and initial level L_{start} and a final level L_{end}, the step function's value is L_{start} for $s < s_0$ and L_{end} for $s \geq s_0$. The amplitude of the step is $L_{\text{end}} - L_{\text{start}}$.

- See Naming Functions for information about the **Function Name** section.
- See Common Settings for the Function Nodes for information about the **Smoothing** section.

Fluid Valve: model library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

Thermodynamics Package

COMSOL provides a **Thermodynamics Package** feature to enable the linking to external physical and thermodynamic property calculations for use in the Chemical Reaction Engineering Module.

For more information see the *Chemical Reaction Engineering Module User's Guide*.

Triangle

A **Triangl**e function (⋀) is a linear increase and linear decline within an interval and `0` everywhere else. You can use the triangle function for band-pass filtering, for example; that is, use it to select values within an interval. The triangle function is a function of one variable (the time t, for example). The default **Function name** is `tri1`.

To add a global or local (for any Component) Triangle function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Triangle** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** (☰) node or the **Definitions** (≡) node and choose it from the **Functions** submenu.

LIMITS

Enter a **Lower limit** (the default is –0.5) and **Upper limit** (the default is 0.5) to specify the interval for the triangle function. For example, if the input argument is time, enter a start and end time. In the midpoint of the interval, this function evaluates to 1, and moving toward the interval boundaries it falls off to 0. Outside the interval it evaluates to 0.

- See Naming Functions for information about the **Function Name** section.
- See Common Settings for the Function Nodes for information about the **Smoothing** section.

Waveform

A **Waveform** function ( ) is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. The waveform function is a function of one variable (the time *t*, for example). The default **Function name** is `wv`.

To add a global or local (for any Component) Waveform function:

- On the **Home** toolbar (Windows users) or **Main** toolbar (Mac and Linux users) select **Waveform** from the **Functions>Global** or **Functions>Local** menu, or
- Right-click the **Global Definitions** ( ) node or the **Definitions** ( ) node and choose it from the **Functions** submenu.

> See Naming Functions for more information about the **Function Name** section.

PARAMETERS

Select a waveform **Type**: **Sawtooth**, **Sine** (the default), **Square**, or **Triangle**. For any selection, enter **Angular frequency** (default is 1), **Phase** (unit: radians; the default is 0), and **Amplitude** (default is 1) values.

> If you have the Batteries & Fuel Cells Module, see 1D Lithium Ion Battery Model for Thermal Models: model library path **Batteries_and_Fuel_Cells_Module/Batteries/li_battery_1d_for_thermal_models**.

Specifying Discontinuous Functions

To specify a discontinuous function, such as a step in space or time, you can use logical functions that evaluate to 1 if true and 0 otherwise. For instance, the following function defines a sine wave that exists for 10 seconds and afterward takes the value 0:

```
sin(2*pi*t)*(t<10)
```

If a coefficient or a material property contains a step function or some other discontinuity, convergence problems can arise. For time-dependent problems, the time-stepping algorithm can run into problems. For stationary problems, mesh-resolution issues can arise such as overshooting or undershooting of the solution due to infinite flux problems. To avoid problems with a discontinuity, replace it with a smoothed step function that emulates steps. Doing so serves two purposes:

- Numerical reliability and convergence are improved.
- What is thought of as a step function is, in reality, a smoothed continuous function because of inertia.

SMOOTHING OF DISCONTINUOUS FUNCTIONS

The easiest way to create a smooth step is to use the predefined Step function. It includes smoothing by default. The Piecewise, Ramp, Rectangle, and Triangle functions also include smoothing (active by default for Rectangle and Triangle functions).

Smoothed Step and Rectangle functions are defined by piecewise 5th-degree polynomials, smoothed Ramp functions by piecewise 4th-degree polynomials, and smoothed Triangle functions by piecewise 3rd-degree polynomials. None of these functions have any overshoot or undershoot.

ADDITIONAL SMOOTHED FUNCTIONS

In addition, the following smoothed functions are available:

- `flsmhs`, a smoothed step function, or *Heaviside function*, with a continuous first derivative and overshoot on both sides of the step. The overshoot ensures that the integral from 0 to infinity is correct. `y=flsmhs(x,scale)` approximates the logical expression `y = (x>0)` by smoothing the transition within the interval $-\text{scale} < x < \text{scale}$. `fldsmhs` is the derivative of the smoothed Heaviside function.
- `flsmsign`, a smoothed sign function with a continuous first derivative. `y = flsmsign(x,scale)` approximates the function `y = sign(x)` by smoothing the transition within the interval $-\text{scale} < x < \text{scale}$. `fldsmsign` is the derivative of the smoothed sign function.
- `flc1hs`, a smoothed Heaviside function with a continuous first derivative without overshoot. Its syntax is similar to the functions just described.
- `flc2hs`, a smoothed Heaviside function with a continuous second derivative without overshoot. Its syntax is similar to the functions just described.

These functions can be useful as a complement and extension to the predefined Step function. In the interval `-scale<x<scale`, the functions `flsmhs` and `flsmsign` are defined by a 7th-degree polynomial chosen so that the 2nd derivative is continuous. Moreover, the moments of order 0, 1, and 2 agree with those for the Heaviside function and the sign function, respectively. This implies that the functions have small overshoots.

Now consider an example. Use `flc1hs` to model the heat capacity C_p of a phase-changing material. Assume that a crystalline material has a heat capacity of 1 kJ/kg. Its melting point at the present pressure is 273.15 K. The liquid phase has a heat capacity of 2 kJ/kg. Create a parameter `scale` with a value of 0.1 and then an **Analytic** node where an analytic function HeatCapacity is defined using the following expression with an argument `T`: `1+flc1hs(T-273,scale)`; then define a plot range of 272.5–273.5 K under **Plot Parameters** and click the **Plot** button in the **Analytic** settings window to plot C_p around the melting point.

Component Couplings

About Component Couplings and Coupling Operators

Component Coupling was previously called Model Coupling.

Component couplings establish couplings between different parts of a model or between different models. A component coupling is defined by a *coupling operator*, taking an expression as its argument. When the operator is used at a point in the destination, the value is computed by evaluating the argument in the source. You can use a coupling operator to compute several quantities that use the same source and destination by calling it with different arguments. All types of component couplings have a source and a destination.

Scalar coupling operators, such as the Integration coupling operator, have a global destination that does not need to be defined; the values are available globally.

The *source* is a subset of a single model (such as some domains or boundaries) where the coupling operator evaluates the supplied expression (as an integration over the source, for example), while the *destination* is the part of the geometry where the result of the coupling operator is defined. The destination can, depending on the type of coupling, be a subset of one or several models or a “global destination” (for a scalar value that is available everywhere). The coupling operator’s value is computed by evaluating the expression given as an argument at one or several points in the source. The source and destination are both geometrical objects, but the source is limited to one geometry whereas the destination is often global.

To add a Component Coupling to any Component:

- On the **Definitions** toolbar select features from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose an option from the **Component Couplings** submenu.

ABOUT COUPLING OPERATORS

Coupling operators are useful for modeling coupled problems and are generalizations of expressions. They are defined by first selecting the source, where the argument of the operator is evaluated, and, in some cases, a destination. An expression to evaluate is not required when you define a coupling operator; instead you can use coupling operators in different modeling contexts, passing the expression to evaluate as an input argument.

There are three categories of coupling operators:

- *Extrusion*. These operators—General Extrusion, Linear Extrusion, Boundary Similarity, and Identity Mapping—connect a source and a destination and take an expression as an argument. When it is evaluated at a point in the destination, its value is computed by evaluating the argument at a corresponding point in the source. When the source and destination are of the same space dimension, it is typically a pointwise mapping. When the destination has a higher dimension than the source, the mapping is done by extruding pointwise values to the higher dimensions. For some examples of the use of extrusion coupling operators, see Examples of Extrusion Couplings.
- *Projection*. These operators—General Projection and Linear Projection—evaluate a series of line or curve integrals on the source, where the line or curve positions depend on the positions of the evaluation points in the destination. In this way it is possible to compute the integral of an expression over one space variable for a range

of different points along the other space axis, giving a result that varies over the latter space variable. For example, you can obtain the average along the y direction of a variable u defined on some 2D domain in the xy-plane by computing the integral

$$\bar{u}(x) = \int u(x, y)\, dy$$

COMSOL uses a method whereby it first applies a one-to-one map to the mesh of the source. It then carries out the integrals in the source over curves that correspond to vertical lines in the transformed source mesh. You can define the map between source and destination in two ways: as a *linear projection* or as a *general projection*. For some examples of the use of projection coupling operators, see Examples of Projection Couplings

- *Scalar*. These operators—Integration, Average, Maximum and Minimum—define a scalar value such as an integration, the average over a set of geometric entities, or the maximum or minimum value of an expression and have a “global destination” (that is, they are available everywhere in the model):
  - An *Integration coupling operator* is the value of an integral of an expression over the source, which is a set of geometric entities (domains, for example).
  - An *Average coupling operator* computes the average of an expression over the source.
  - A *Maximum* or *Minimum coupling operator* computes the maximum or minimum, respectively, of an expression over the source.

  These operators can be evaluated anywhere in a model, and the value does not depend on where in the model the evaluation occurs. Integration couplings are useful for evaluating integrated quantities. To evaluate the total current across a boundary in a 2D Electric Currents model, for example, define an integration coupling operator `intop1` with a source on the boundary where the current flows. Then the value of `intop1(ec.normJ*ec.d)`, where `normJ` is the current density norm (SI unit: A/m^2) and `d` is the thickness of the 2D geometry (SI unit: m), is the total current flowing across that boundary (SI unit: A). For some other examples of the use of integration coupling operators, see Examples of Integration Couplings

Coupling operators can:

- Make the value and the exact Jacobian of an expression available nonlocally.
- Take information from a boundary, for example, and make it available on other parts of a model (a domain, for example).
- Be used for results evaluation and visualization purposes.
- Define nonlocal couplings including mesh transformations, integrals over domains and boundaries, and projections.

About Source and Destination Mappings

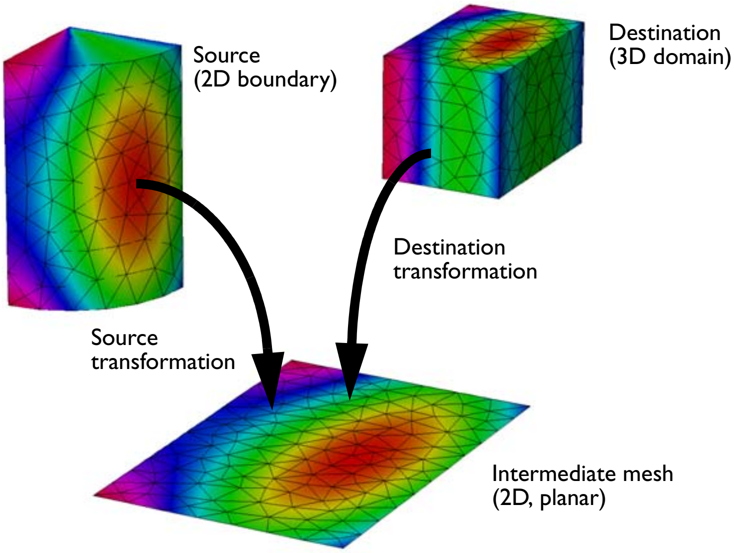


Figure 5-1: An example of a general extrusion mapping.

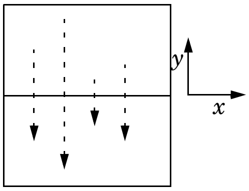
The definition of any extrusion component coupling involves two mesh maps. The *source map* is a one-to-one mapping that maps the mesh of the physical source of dimension srcedim to an *intermediate mesh* of the same dimension embedded in a space of dimension idim ≥ srcedim. The *destination map* is a mapping from the destination of dimension dstedim, where the operator can be evaluated, to the same space that contains the intermediate mesh.

When the value of the coupling operator is requested somewhere in the destination, the software transforms the destination points using the destination map. It compares the resulting coordinates to the elements in the intermediate mesh to find corresponding locations in the physical source. This means that the source map must be inverted but not the destination map. The latter can in fact be noninvertible, which is, for example, the case when dstedim > idim, leading to an extrusion.

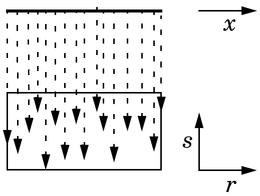
To avoid the need to solve a nonlinear system of equations for every destination point, COMSOL assumes that the source map is linear on each element of the intermediate mesh. In practice, the map is often trivial and leaves the coordinates unchanged, but it can also rescale, stretch, bend, or reflect the mesh.

Examples of Extrusion Couplings

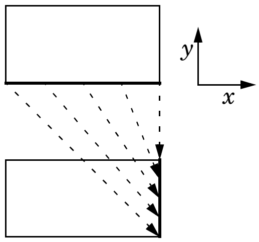
All the graphics in these examples use **General Extrusion** component coupling.



One application of a **General Extrusion** coupling is to mirror the solution on the *x*-axis. This can be useful for analysis. Both source and destination are two-dimensional, as well as the intermediate mesh (srcedim = idim = dstedim). The source map to enter is x, y, and the destination map is x, -y. This can also be done with **Linear Extrusion**.



Another **General Extrusion** example is to extrude the solution in the 1D geometry to a 2D domain along the s-axis. The source map is `x`, and the destination map is `r`, so here srcedim = idim = 1, dstedim = 2.

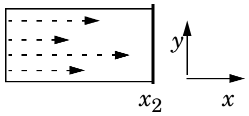


Another example maps values on the lower boundary of a rectangle that extends from $x = -1$ to $x = 1$ and from $y = 0$ to $y = 1$, to the right boundary on the same rectangle. The source map is `(x+1)/2` and the destination map is `y`. Both maps have a single component since srcedim = idim = dstedim. This map is also linear and can be done with **General** or **Linear Extrusion**, or with **Boundary Similarity**.

Extrusion Component Coupling—Example 2

Consider the case of a single rectangular domain where the source term in Poisson's equation comes from the inward flux over the right boundary for the corresponding y coordinate.

$$\begin{aligned} -\Delta u &= \frac{\partial}{\partial n} u(x_2, y) \quad \text{on } \Omega \\ u &= xy \quad \text{on } \partial\Omega \end{aligned}$$

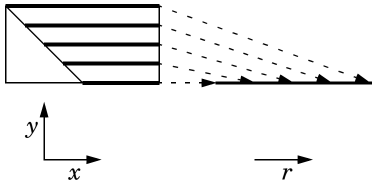


The figure to the left illustrates the extrusion process. The values of the influx on the boundary become available throughout the domain by extrusion along the y-axis. The source map is `y`, and the destination map is `y`.

EXAMPLES OF PROJECTION COUPLINGS

All the graphics in these examples use the **General Projection** component coupling. These examples can also be done using **Linear Projection**.

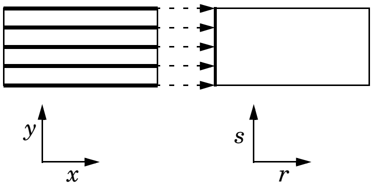
Projection Coupling—Example 1



For each point r, the coupling operator returns the integral

$$v(r) = \int_{\substack{y = r/2 \\ (x, y) \in S_2}} u(x, y) dx$$

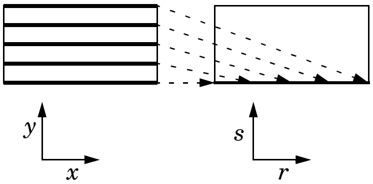
The source map is `y, x`, and the destination map is `r/2`.



For each point $(0, s)$, the coupling operator returns the integral

$$v(0,s) = \int_{\substack{y=s \\ (x,y)\in S_2}} u(x,y)dx$$

The source map is `y`, `x`, and the destination map is `s`.



For each point $(r, 0)$, the coupling operator returns the integral

$$v(r,0) = \int_{\substack{y=r/2 \\ (x,y)\in S_2}} u(x,y)dx$$

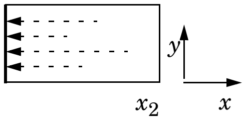
The source map is `y`, `x`, and the destination map is `r/2`.

The integration can also sweep nonrectangular domains. The integrals include only the source domains; they exclude other domains and the external area.

Projection Coupling—Example 2

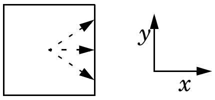
Consider the case of a single rectangular domain with Poisson's equation. Integrate the solution squared along lines parallel to the x-axis and make the result available for analysis on the left boundary.

$$\begin{aligned} -\Delta u &= 1 \quad \text{on } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$



The figure illustrates the projection process. Project the integral of the solution squared on the boundary. The source map is `y`, `x` and the destination map is `y`. If the projection operator is called `genproj1`, the desired result is obtained by evaluating `genproj1(u^2)`.

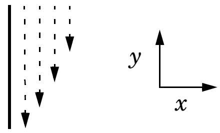
EXAMPLES OF INTEGRATION COUPLINGS



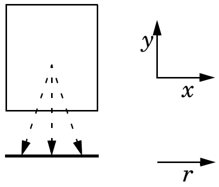
Consider Poisson's equation on a rectangular domain. The integral of the solution squared serves as the influx in a Neumann boundary condition on the right boundary. There is a Dirichlet boundary condition on the left boundary, and the top and bottom boundaries have zero influx.

$$\begin{aligned} -\Delta u &= 1 && \text{on } \Omega \\ u &= x && \text{on } \partial\Omega_1 \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega_{2,3} \\ \frac{\partial u}{\partial n} &= -\int_\Omega u^2 d\Omega && \text{on } \partial\Omega_4 \end{aligned}$$

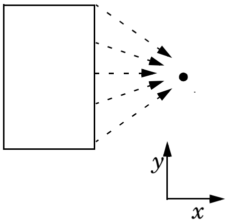
For example, define an integration coupling operator called `intop1`, with the rectangular domain as source. The influx for the Neumann boundary condition is obtained as `intop1(u^2)`.



A second example is when a scalar value from a vertex is used everywhere on a boundary to which the vertex belongs. In structural mechanics use this type of coupling to formulate displacement constraints along a boundary in terms of the displacements of the end point. In electromagnetics the same technique can implement *floating contacts*.



Another example is to use the integral over a domain in a 2D geometry along a domain in another 1D geometry. This approach is helpful for process-industry models where two processes interact.



Integration coupling operators can implement *integral constraints*. First define a coupling operator at some vertex in such a way that it represents the value of the integral to be constrained. Then use a point constraint to set the coupling operator, and thereby the integral, to the desired value.

NONLOCAL COUPLINGS AND THE SPARSITY OF THE JACOBIAN

The Jacobian for problems formulated using the finite element method is usually large but sparse. This is because the solution at each mesh node can depend at most on the degrees of freedom at the neighboring mesh elements. However, by introducing nonlocal component couplings using coupling operators, nonlocal dependencies are created that fill up the rows and columns of the affected source and destination nodes. These additional elements might make the Jacobian matrix only slightly less sparse, with marginal effects on solution speed; they can also make it a great deal less sparse, in which case memory use and CPU time involved in solving the problem increases considerably. For this reason, take particular care when introducing nonlocal couplings. For example, defining a heat source based on an integration operator over the whole domain that is also a function of temperature (the dependent variable) leads to a coupling between all the degrees of freedom in the model creating a full Jacobian matrix instead of the sparse limited-bandwidth matrices typical of finite element models.

You can prevent the fill-in of the Jacobian matrix using the `nojac` operator, which forces COMSOL to exclude the expression that it encloses when forming the Jacobian. Using the `nojac` operator can slow down the convergence of the solution. Another possible solution is to add a single degree of freedom that represents the value of an expression with a scalar coupling operator.

General Extrusion

A **General Extrusion** coupling operator () maps an expression defined on a source to an expression that can be evaluated on any destination geometry where the destination map expressions are valid. Compared to the *linear extrusion*, these operators define a more general, possibly nonlinear, relation between source and destination.

Specifically, when the destination has more space dimensions than the source, the operator performs extrusion of values. The default **Operator name** is `genext1`.

To add a General Extrusion Component Coupling for any Component:

- On the **Definitions** toolbar select **General Extrusion** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, **Source**, and **Advanced** sections.

DESTINATION MAP

Specify the general extrusion destination map by entering an expression in the **x-expression**, **y-expression**, and **z-expression** fields. This maps each point in the destination to a point in the intermediate mesh, where the argument of the extrusion operator is evaluated. A general extrusion operator can be evaluated at any point where the destination map expressions are defined.

The number of destination map expressions is the same as the space dimension of the intermediate mesh. For example, if the intermediate mesh is in two-dimensional space, there is no **z-expression** field.

If you have the:

- Acoustics Module, see Flow Duct: model library path **Acoustics_Module/Industrial_Models/flow_duct**.
- Chemical Reaction Engineering Module, see Packed Bed Reactor: model library path **Chemical_Reaction_Engineering_Module/Packed_Bed_Reactors/packed_bed_reactor**.
- Geomechanics Module, see Concrete Beam With Reinforcement Bars: model library path **Geomechanics_Module/Tutorial_Models/concrete_beam**.
- RF Module, see Radar Cross Section: model library path **RF_Module/Scattering_and_RCS/radar_cross_section**.
- Subsurface Flow Module, see Aquifer Characterization: model library path **Subsurface_Flow_Module/Fluid_Flow/aquifer_characterization**.

Linear Extrusion

A **Linear Extrusion** coupling operator ( ) maps an expression defined on a source to an expression that can be evaluated in the destination. Use this to define a linear mapping of this kind. Linear extrusion can be used when the correspondence between evaluation points in the source and destination is linear and in some nonlinear cases. Otherwise, use a general extrusion coupling. The Linear Extrusion operator defines a linear extrusion that maps between geometric parts of the same dimension. The parts can exist in geometries of different space dimensions. For example, you can couple edges (boundaries) in 2D to edges in 3D; or couple 2D domains to 3D faces. In these cases geometries of different space dimensions are needed for the source and destination. You define the linear extrusion by specifying points in both source and destination. The default **Operator name** is `linext1`.

To add a Linear Extrusion Component Coupling for any Component:

- On the **Definitions** toolbar select **Linear Extrusion** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

> Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, **Source**, **Source Vertices**, **Destination Vertices**, and **Advanced**, sections.

DESTINATION

The mapping from destination to source is defined as the following:

- First, the destination is orthogonally projected onto the linear space spanned by the destination vertices.
- Then this linear space is mapped linearly to the source, so that each destination vertex is mapped to the corresponding source vertex.

Select an option from the **Destination geometry** list if there is more than one geometry in the model. A linear extrusion operator can only be evaluated on the destination geometry and the destination vertices must be chosen in the destination geometry. Select an option from the **Destination frame** to evaluate the destination vertex coordinates in the specified frame.

> - If you have the CFD Module, see Turbulent Flow Over a Backward Facing Step: model library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**.
> - If you have the Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: model library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**.

Boundary Similarity

The **Boundary Similarity** coupling operator ( ) maps an expression defined on a part of a boundary to another part of a boundary with the same shape. This operator is slightly different for 2D and 3D models:

- In 3D, the destination map is a similarity that maps a destination boundary onto a set of source boundaries. The mesh is always viewed in the mesh frame. By default, the algorithm automatically chooses a map when symmetries make several maps possible. To control this choice in 3D, add a One-Point Map, Two-Point Map, or Edge Map subnode ( ).
  - *Edge Map:* Specify that a certain destination edge should be mapped onto a certain source edge. The edge's relative direction is given by the property direction. The edges must be adjacent to the given boundary.
  - *One-Point Map:* Specify that a certain destination vertex should be mapped onto a certain source vertex.
  - *Two-Point Map:* Specify that two destination vertices should be mapped onto two source vertices.
- In 2D, it works the same except the destination map is a similarity that maps a destination edge onto a set of source edges and there are no subnodes to add.

The default **Operator name** is `bndsim1`.

To add a Boundary Similarity Component Coupling for any Component:

- On the **Definitions** toolbar select **Boundary Similarity** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Boundaries** and **Destination Boundary** sections, as well as the rest of the **Advanced**, section.

ADVANCED

For 2D models, select a relative **Direction** of the source and destination edges—**Automatic orientation** (the default), **Same orientation**, or **Opposite orientation**. This specifies the relative direction of the source and boundary edges and specifies which one of the two possible similarity mappings between the source and destination edge to use.

For 3D models, a similar direction functionality is provided by the subnodes One-Point Map, Two-Point Map, and Edge Map to exactly specify the similarity mapping between the source and destination when more than one possibility exists.

Select the **Use source map** check box to have a nonlinear correspondence between the source and destination. The source map is specified by entering expressions in the **x-expression**, **y-expression**, and **z-expression** fields.

Only one map node is allowed per boundary similarity coupling and only one source or destination point per field is allowed.

One-Point Map

Use a **One-Point Map** to control a Boundary Similarity component coupling mapping in 3D. Right-click the Boundary Similarity node to add a One-Point Map subnode.

POINTS

To select a single **Point on source** and a single **Point on destination** adjacent to the source and destination selection, click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination boundaries. Select **Manual** or **All points** for the points on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add points to the **Selection** section. If required, click the **Swap source and destination** button (↕).

Two-Point Map

Use a **Two-Point Map** to control a Boundary Similarity component coupling mapping in 3D. Right-click the Boundary Similarity node to add a One-Point Map subnode.

SOURCE POINTS

To select a single **First point on source** and a single **Second point on source**, click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination points. Select **Manual** or **All points** for the points on the source. If **Manual** is selected, click in the **Graphics** window to add points to the **Selection** section. If required, click the **Swap Source and Destination** button (↕).

DESTINATION POINTS

To select a single **First point on destination** and a single **Second point on destination** adjacent to the destination selection, click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination points. Select **Manual** or **All points** for the points on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add points to the **Selection** section. If required, click the **Swap Source and Destination** button (↕).

Edge Map

Use an **Edge Map** to control a Boundary Similarity component coupling mapping in 3D. Right-click the Boundary Similarity node to add a One-Point Map subnode.

EDGES

To select a single **Source edge** and a single **Destination edge**, click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination edges. Select **Manual** or **All edges** for the edges on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add edges to the **Selection** section. If required, click the **Swap Source and Destination** button (↕). The destination edge is mapped to the source edge by the similarity mapping from destination to source.

ADVANCED

Select a relative **Direction** of the source and destination edges—**Automatic orientation** (the default), **Same orientation**, or **Opposite orientation**. This specifies the relative direction of the source and boundary edges and specifies which one of the two possible similarity mappings between the source and destination edge to use.

Identity Mapping

An **Identity Mapping** component coupling (⇹) maps between geometric entities that overlap, possibly when viewed in different frames. When it is evaluated at a specific set of coordinates in the destination frame, its argument is evaluated with the same coordinates in the source frame. The default **Operator name** is `idmap1`.

To add an Identity Mapping Component Coupling for any Component:

- On the **Definitions** toolbar select **Identity Mapping** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

FRAMES

Select a **Source frame** to use on the source geometric entity and a **Destination frame** to use on the destination geometric entity. In most cases the default **Spatial (x,y,z)** frame can be used in both cases. Or choose **Material**,

Geometry, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics. Choose a **Destination geometry** from the list.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, and **Advanced** sections.

If you have the CFD Module, see Solar Panel in Periodic Flow: model library path **CFD_Module/Single-Phase_Tutorials/solar_panel**.

General Projection

Use a **General Projection** component coupling ( ) to define integration along curves. A projection coupling operator evaluates an expression defined on a source by integration along lines or curves depending on the evaluation point in the destination. The General Projection operator is defined by mapping the source to an abstract intermediate space of dimension srcedim, and the destination to the subspace of dimension srcedim−1 obtained by setting the last coordinate to 0 (srcedim is the dimension of the source selection). To every point in the destination, there is a vertical line in the intermediate space, obtained by allowing the last coordinate to vary while the remaining coordinates are given by the destination map. The set of points in the source selection that are mapped onto this line by the source map is a line or curve, and the projection operator is evaluated by integrating along this line or curve. The default **Operator name** is `genproj1`.

It is only possible to use projection component coupling with simplex elements such as triangles and tetrahedra.

To add a General Projection Component Coupling for any Component:

- On the **Definitions** toolbar select **General Projection** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Source Selection** section.

SOURCE MAP

Specify the general projection source map by entering expressions in the **x-expression**, **y-expression**, and **z-expression** fields.

The dimension of the intermediate space equals the dimension srcedim of the source. If the selection has lower dimension than the source geometry, specify only the first srcedim expressions.

Use expressions containing spatial coordinates in the source geometry when defining the map. The map must be approximately linear within each mesh element. Select a **Source frame** from the list.

DESTINATION MAP

Enter an **x-expression** and, depending on the dimensions, **y-expression** for each coordinate except the last in the intermediate space.

The destination map has one field less than the source map. When defining the map you can use expressions containing spatial coordinates in the destination geometry. The destination mapping can be highly nonlinear or noninvertible.

- If the selection has lower dimension than the source geometry, specify only the first srcedim−1 expressions. A general projection operator can be evaluated at any point where its destination map is defined.
- If the source selection has dimension 1, no destination map needs to be specified, and consequently this section is not shown if the source geometry is 1D. In this case, it is probably better to use an integration coupling.

ADVANCED

Enter an **Integration order** of the numerical integration method (default: 4).

Linear Projection

Use a **Linear Projection** component coupling ( ) when the argument is to be integrated along a line, and the line depends linearly on the evaluation point.

The linear projection maps between a source and a destination of the nearest lower dimension. The source and destination can exist in geometries of different space dimensions. For example, you can couple domains in 2D to edges in 3D or couple 3D domains to 2D domains. You define the linear projection by specifying points in both the source and destination. The default **Operator name** is `linproj1`.

It is only possible to use projection component coupling with simplex elements such as triangles and tetrahedra.

To add a Linear Projection Component Coupling for any Component:

- On the **Definitions** toolbar select **Linear Projection** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, **Source Vertices**, and **Destination Vertices** sections.

SOURCE

Select a **Source frame** from the list to evaluate the coordinates of the source vertices in the selected frame.

Then specify the linear projection by giving a set of points in the source and in the destination. The order of the vertices is significant. COMSOL constructs a linear projection from the source to the destination using the subspaces spanned by the vertices. Denote the map rank by n, denote the source vertices by $x_0, x_1,..., x_n$, and denote the destination vertices $x'_0, x'_1,..., x'_n$. After padding the source and destination vertices' vectors with zeros as necessary, the software solves the following matrix equation for a *transformation matrix* T and a *translation vector* V:

$$
\begin{aligned}
x'_0 &= Tx_0 + V \\
x'_1 - x'_0 &= T(x_1 - x_0) \\
&\dots \\
x'_n - x'_0 &= T(x_n - x_0)
\end{aligned}
$$

For the projection component coupling there must be one more vertex in the source than in the destination.

DESTINATION

Select an option from the **Destination geometry** list if there is more than one geometry in the model.

- A linear projection operator can only be evaluated on the destination geometry and the destination vertices must be chosen in the destination geometry.
- The destination vertex coordinates are evaluated in the selected **Destination frame**.

ADVANCED

Enter an **Integration order** of the numerical integration method (default: 4).

Integration

An **Integration** component coupling (∫du) integrates an expression over the source (some selected geometric entities like domains, boundaries, or edges). You can also use it with a point as the source to make the value of an expression at that point available globally. The integral is evaluated by integrating the expression (integrand) in the argument over the source (or, in some cases, by summing the expression over the node points in the source). Integration coupling operators have global destination, so they can be evaluated anywhere in the model. Because it is an operator, you can define one integration operator (`intop1`, for example) for a part of the geometry (a boundary, for example) and then use that several times in the model to compute integrals over that boundary for different integrands. For example, `intop1(T)` is the integral of the temperature T over the boundary, and `intop1(1)` is simply the length (2D) or area (3D) of the boundary. Also, using the `dest` operator it is possible to create convolution integrals. The default **Operator name** is `intop1`.

To add an Integration Component Coupling for any Component:

- On the **Definitions** toolbar select **Integration** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, and **Advanced** sections.

- Acoustics of a Muffler: model library path: **COMSOL_Multiphysics/Acoustics/automotive_muffler**
- Fluid Valve: model library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

Average

An **Average** component coupling (AV) computes the average of an expression over the source (some selected geometric entities). It can be evaluated anywhere in any model. It is similar to the Integration operator; the difference being that the integral is divided by the volume, area, arc length, or number of points in the source, depending on the type of geometric entities in the source. The default **Operator name** is `aveop1`.

To add an Average Component Coupling for any Component:

- On the **Definitions** toolbar select **Average** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

Go to Common Settings for Component Couplings for information about the **Operator Name**, **Source Selection**, and **Advanced** sections.

Effective Diffusivity in Porous Materials: model library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**

Maximum and Minimum

The **Maximum** (MAX) and **Minimum** (MIN) coupling operators compute the maximum or minimum of an expression over selected geometric entities in the source and gives the maximum or minimum value of the expression in the argument over the source. The operator can be evaluated anywhere in any model. Two arguments can be given, and the returned value is then the value of the second argument evaluated in the max/min of the first argument. This is useful for evaluating, for example, the location of the maximum or minimum. In a 2D model where the temperature T is solved for, use the following syntax for the maximum operator `maxop1` in a **Global Evaluation** node, for example, to get the x- and y-coordinate for the maximum of the temperature: `maxop1(T,x)`and `maxop1(T,y)`. The default **Operator name** is `maxop1` or `minop1`.

When a **Max/Min Volume**, **Max/Min Surface**, or **Max/Min Line** plot is used, the maximum and minimum values, along with the coordinates for the corresponding locations, appear in a table (underneath the plot with the default COMSOL Desktop layout).

To add a Maximum or Minimum Component Coupling for any Component:

- On the **Definitions** toolbar select **Maximum** or **Minimum** from the **Component Coupling** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Component Couplings** submenu.

ADVANCED

Select a **Point type**—**Node points** (the default), **Integration points**, or **Lagrange points**. The point type controls the choice of evaluation points—the result is more accurate with more points, but more points also means a slower evaluation.

- If **Integration points** is selected, enter an **Integration order**. The default is `4`.
- Select **Lagrange points** to compute the maximum or minimum by evaluating the expression in the argument at a finite set of points in the source and taking the maximum or minimum of these values. If it is selected, enter a **Lagrange order**. The default is `2`.

Go to Common Settings for Component Couplings for information about the **Operator Name** and **Source Selection** sections.

Deformation of a Feeder Clamp: model library path **COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp**

Common Settings for Component Couplings

The following sections in the settings windows for the component coupling nodes are similar or the same for some of the component coupling nodes and are described in this section.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name. This is the name that is used to access the operator in the model, so use a name that describes it well.

SOURCE SELECTION

The source selection defines the source for the component coupling—the part of the geometry where the coupling operator evaluates the supplied expressions.

From the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list. If **Manual** is selected, select geometric entities in the **Graphics** window. Select **All domains**, for example, to add all applicable geometry to the **Selection** list.

SOURCE VERTICES AND DESTINATION VERTICES

The selection of **Source Vertices** and **Destination Vertices** define the linear mapping from the destination to the source.

Click the **Active** button activate one of vertex selections. For Windows users, the buttons are ON □ and □ OFF. For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Select a single source vertex for each of **Source vertex 1**, **Source vertex 2**, **Source vertex 3**, and **Source vertex 4**. Then select a single destination vertex for each of **Destination vertex 1**, **Destination vertex 2**, **Destination vertex 3**, and **Destination vertex 4** (vertex 4 is available for Linear Extrusion only).

- For Linear Extrusion: The number of source vertices must be at least one and not more than 1+min(srcsdim,dstsdim), where srcsdim and dstsdim are the dimensions of the source and destination geometries, respectively. The number of destination vertices entered should be the same as the number of source vertices. If not all destination vertex selections are used, the empty selections must be last.
- For Linear Projection, select srcedim+1 source vertices where srcedim is the dimension of the source selection. Depending on the dimension of the source selection, it can be that some of the last source vertex selections should be left empty. The number of destination vertices should be *one less* than the number of source vertices. If not all destination vertex selections are used, the empty selections must be last. Select srcedim destination vertices where srcedim is the dimension of the source selection. Depending on the dimension of the source selection, it can be that some of the last destination vertex selections should be left empty.

  An evaluation point in the destination geometry is first orthogonally projected onto the linear space spanned by the destination vertices (unless they span the entire space). The projected point is then mapped to the source geometry by a linear mapping taking each destination vertex to the corresponding source vertex. Let L be the line through this point, which is parallel to a line through the first and the last source vertex. If the source selection lies in the linear space spanned by the source vertices, the Linear Projection operator is evaluated by integrating along L. In general the operator is evaluated by integrating along the line or curve in the source selection, which is mapped to L under orthogonal projection onto the linear space spanned by the source vertices.

SOURCE BOUNDARIES AND DESTINATION BOUNDARY

Select **Manual** or **All boundaries** from the **Selection** list to define the source selection. If **Manual** is selected, select boundaries in the **Graphics** window. Select **All boundaries** to add all boundaries to the **Selection** list.

There can only be one destination boundary. Click the **Active** button to enable or disable the **Destination Boundary** selection. For Windows users, the buttons are ON □ and □ OFF. For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF. Then choose the boundary in the **Graphics** window.

SOURCE FRAME AND SOURCE MAP

Select a **Source frame** to use in the source. In most cases the **Source** section default settings can be used. Optionally, select the **Use source map** check box and enter expressions in the **x-expression**, **y-expression**, and **z-expression** fields for the source map from the source to the intermediate mesh.

For the General Extrusion component coupling, the number of source map expressions is the same as the number of destination map expressions. With the default source map expressions, the intermediate mesh can be considered identical to the source.

The dimensionality idim of the intermediate space is determined by the number of nonempty source and destination map expressions, which must be the same, and must also satisfy srcedim ≤ idim ≤ srcsdim, where srcedim is the dimension of the source selection, and srcsdim is the dimension of the source geometry.

ADVANCED SETTINGS FOR COMPONENT COUPLINGS

For the General Extrusion, Linear Extrusion, Boundary Similarity and Identity Mapping component couplings, select an option from the **Mesh search method** list to specify what should happen if an evaluation point in the destination is mapped to a point outside the source:

- If **Use tolerance** is selected (the default) the result depends on the other field definitions in this section.
- If **Closest point** is selected, the closest point in the source selection is used.

Enter a value in the **Extrapolation tolerance** field. If the mapped point is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be in the source. Otherwise, the mapping fails.

Select the **Use NaN when mapping fails** check box to evaluate the operator to NaN (Not-a-Number) if the mapping fails. Otherwise an error occurs.

For the Integration and Average couplings, select **Integration** or **Summation over nodes** from the **Method** list. In most cases use integration. Summation over nodes is useful, for example, for calculating reaction forces. If **Integration** is selected, enter a value in the **Integration order** field.

Also, when working with multiple frames, select a **Frame** from the list for the volume element to be used in the integration.

For axisymmetric geometries, select the **Compute integral in revolved geometry** check box to perform the integration in 3D (for a 2D axisymmetric model) or in 2D (for a 1D axisymmetric model).

Coordinate Systems

About Coordinate Systems

COMSOL uses a global Cartesian coordinate system by default to specify material properties, loads, and constraints in all physics interfaces and on all geometric entity levels (points, edges, boundaries, and domains). In boundary conditions and fluid domains, the global system is generally interpreted as having fixed axis directions in space; that is, it is a *spatial frame* system. When specifying properties of solid materials, the global system axes are instead fixed in the material. In other words, it is a *material frame* system in that context.

Not only the global coordinate system, but also coordinate systems defined as a rotation relative to the global system, are context-dependent in this way. Such systems are collectively referred to as *relative coordinate systems*, to distinguish them from *absolute coordinate systems*.

The spatial Cartesian coordinate system coordinates default to the following names in 2D and 3D (in 2D axisymmetric geometries, COMSOL uses cylindrical coordinates):

GEOMETRY	DEFAULT NAME OF SPATIAL COORDINATES
2D	*x y*
3D	*x y z*
Axial symmetry 2D	*r* φ *z*

In 3D, an image displays in the lower-left corner of the **Graphics** window to indicate the orientation of the global coordinate system.

User-defined coordinate systems can be used on all geometric entity levels to simplify the modeling process. In some of the physics interfaces, these coordinate systems can be used to define orthotropic and anisotropic material properties that are not aligned with the global Cartesian coordinate system. See Table 5-15 for an overview of the available coordinate systems. Note in particular that some coordinate systems specify absolute directions in space, while others specify a rotation relative to the default global system, as indicated by the Type column in the table.

To add a Coordinate System to any Component:

- On the **Definitions** toolbar select features from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose an option from the **Coordinate Systems** submenu. :

TABLE 5-15: COORDINATE SYSTEM DESCRIPTIONS

NAME AND LINK	ICON	TYPE	DESCRIPTION
Base Vector System		relative	1D, 2D, and 3D. Define this using a set of base vectors to form a coordinate system, which are declared as orthonormal.
Boundary System		absolute	2D and 3D. A local base vector system on 2D boundaries (t, n) and on 3D boundaries (t1, t2, n). Use it to apply loads that apply in a normal or tangential direction on a boundary that is not aligned with the global Cartesian coordinate system. This coordinate system is always available.
Cylindrical System		absolute	2D and 3D. Use this where rotational symmetry about the axis is required. Not applicable in geometries with 2D axial symmetry, where a cylindrical coordinate system is the default coordinate system.

TABLE 5-15: COORDINATE SYSTEM DESCRIPTIONS

NAME AND LINK	ICON	TYPE	DESCRIPTION
Mapped System		absolute	1D, 2D, and 3D. This can deal with translated and rotated coordinate systems. Use this to create a system that defines a mapping from the frame coordinate system.
Rotated System		relative	2D and 3D. Use this to define rotation about the out-of-plane direction in 2D and Euler angles in 3D.
Spherical System		absolute	3D only. Use this when a field or property using spherical coordinates is to be specified.
Scaling System		absolute	For physics that support infinite elements or perfectly matched layers only. Use this coordinate system, which is similar to a mapped coordinate system, to arbitrarily deform the domain.

- Grouping Nodes by Space Dimension and Type
- Spatial Coordinate Variables

COORDINATE SYSTEM IDENTIFIER

The **Coordinate System Identifier** provides a namespace for variables created by the Coordinate system. The default identifier is `sys1`. For example, the determinant of the coordinate system's transformation matrix can typically be accessed in equations and postprocessing as `sys1.detT`. See the **Equation View** subnode for a complete list of available variables.

To display the **Equation View** node under all nodes creating variables, click the **Show** button and select **Equation View**. See also Equation View.

Base Vector System

Define a **Base Vector System** ( ) using a set of base vectors to form a coordinate system. The system does not necessarily need to be orthonormal, but when it is, declaring it orthonormal and linear enables simplifications which improve performance.

A vector $\mathbf{F}$ is represented by its contravariant components $[F_1, F_2, F_3]^T$ in the base of the new base vector system defined by the base vectors $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ on the form $\mathbf{F} = F_1\mathbf{u}_1 + F_2\mathbf{u}_2 + F_3\mathbf{u}_3$. Expressing the base vectors as components in another system (for example, the global spatial system $[\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z]$) gives the transformation matrix between bases:

$$\begin{bmatrix}F_x\\F_y\\F_z\end{bmatrix} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}\cdot\begin{bmatrix}F_1\\F_2\\F_3\end{bmatrix}$$

$$\begin{bmatrix}F_1\\F_2\\F_3\end{bmatrix} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}^{-1}\cdot\begin{bmatrix}F_x\\F_y\\F_z\end{bmatrix} = \left\{\begin{matrix}|\mathbf{u}_i| = 1\\ \mathbf{u}_i\cdot\mathbf{u}_j = \delta_{ij}\end{matrix}\right\} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}^{T}\cdot\begin{bmatrix}F_x\\F_y\\F_z\end{bmatrix}$$

where the last equality holds when the base vector system is orthonormal.

Note that you specify the base vectors as components in the default global coordinate system, which is context-dependent. The base vector system is therefore a relative coordinate system whose interpretation depends on the interpretation of the global system in the current context.

The Curvilinear Coordinates interface can create special base vector systems in **Curvilinear System** nodes ( ). See Curvilinear Coordinates.

To add a Base Vector System to any Component:

- On the **Definitions** toolbar select **Base Vector System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

Coordinate Names

In the **Coordinate names** table, the default names are entered—`x1`, `x2`, and `x3`. In planar 2D models, `x1` and `x2` are typically the in-plane coordinates, and `x3` is the out-of-plane coordinate. Note that these coordinate names are only used as indices for vector and tensor variable names, and cannot be evaluated as variables. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Base Vectors

Define the **Base vectors** in terms of the global Cartesian coordinates (typically `x`, `y`, and z); one base vector on each row (two for 2D and three for 3D).

For 1D models, select which basis vector is parallel to the 1D geometry**.** Select an option from the **In-plane index** list. The default is 1.

For 2D models, select which basis vector to compute as the cross product of the two in-plane vectors specified. Select an option from the **Out-of-plane index** list. The defaults are **3** for a plane 2D model and **2** for an axisymmetric 2D model. For example, to map the first vector, `x1`, to the direction defined by $y = x$ in 2D, enter `1` in the fields under **x** and **y** on the **x1** row.

Simplifications

Set base vector system properties that help simplify the coordinate transformations. Select the **Assume orthonormal** check box if the coordinate system is orthonormal.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

- If you have the Nonlinear Structural Materials Module, see Pressurized Orthotropic Container: model library path **Nonlinear_Structural_Materials_Module/Plasticity/orthotropic_container**.
- If you have the Structural Mechanics Module, see Piezoelectric Shear-Actuated Beam: model library path **Structural_Mechanics_Module/Piezoelectric_Effects/shear_bender**.

Boundary System

A **Boundary System** ( ) is a local base vector system on 2D boundaries ($\mathbf{t}$, $\mathbf{n}$) and on 3D boundaries ($\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$). Use it to apply loads and other boundary conditions in a normal or tangential direction on a boundary that is not aligned with the global Cartesian coordinate system.

For 3D and 2D models, a **Boundary System** node is automatically added under **Definitions**.

Common applications for this coordinate system include specifying pressure or normal displacement on a surface.

To specify the boundary coordinate system, you specify the direction of the normal and a direction that is projected onto the boundary, normalized, and used as the first tangent vector. The normal direction is in most cases the outward-pointing normal vector, but you can reverse the normal direction. The general definition of the normal is the direction of the normal vector $\mathbf{n}$, which can be plotted using the variables for its components (typically `nx`, `ny`, and `nz`). See Normal Variables.

- In 2D the local coordinate system is defined by ($\mathbf{t}_1$, $\mathbf{n}$, $\mathbf{t}_o$), representing the tangential and normal direction of the boundary. This coordinate system is always right-oriented. The second tangent direction ($\mathbf{t}_o$) is the cross product between normal vector ($\mathbf{n}$) and the first tangent direction ($\mathbf{t}_1$). This method always gives a right-oriented orthonormal system, unless the tangent direction is parallel to the normal.
- In 3D the local coordinate system is defined by ($\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$), representing two tangential directions ($\mathbf{t}_1$ and $\mathbf{t}_2$) and one normal direction ($\mathbf{n}$).This coordinate system is always right-oriented but not always orthogonal. The second tangent direction ($\mathbf{t}_2$) is the cross product between the specified normal vector ($\mathbf{n}$) and the first tangent vector ($\mathbf{t}_1$). This method always gives a right-oriented orthonormal system, unless the tangent direction is parallel to the normal.

To add a Boundary System to any Component:

- On the **Definitions** toolbar select **Boundary System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** ( ) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

Frame

Select a **Frame—Deformed configuration** (the default), **Geometry configuration**, or **Reference configuration**. The deformed configuration follows the material whereas the reference configuration is attached to the spatial frame. The geometry configuration is used to specify normal and tangential components of boundary conditions and refers to the undeformed geometry when using a Deformed Geometry interface.

Coordinate Names

In the **Coordinate names** table, the default names are entered—`t1`, `t2`, and `n` (for 3D models) or `t1`, `n`, and `to` (for 2D models). Click the table cells to edit the names. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses. To reverse the direction of the normal for the boundary system, select the **Reverse normal direction** check box.

Select an option from the **Create first tangential direction from** list: **Global Cartesian** (the default) or **Manual**. If **Global Cartesian** is selected, select **1**, **2**, or **3** (that is, *x*, *y*, or *z*) from the **Axis** list. If **Manual** is selected, default values are displayed for the local tangent variables `t1x`, `t1y`, and `t1z` (3D) or `t1x` and `t1y` (2D). Enter other values as required to define a tangent direction by specifying directions for a local tangent plane in the **x**, **y**, and **z** fields.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

Many model examples use this coordinate system. For one example, see Electric Sensor: model library path **COMSOL_Multiphysics/Electromagnetics/electric_sensor**.

Cylindrical System

A **Cylindrical System** ( ) can be used in 2D and 3D where rotational symmetry about the axis is required. The cylindrical coordinate system is not applicable in geometries with 2D axial symmetry. The local coordinate system is defined by (r, φ , a), where r represents the radial distance from the longitudinal axis, φ is the azimuthal angle (in the interval from $-\pi$ to π), and a is the distance from the origin along the longitudinal axis. In 2D models, only the origin can be specified, whereas in 3D models, the longitudinal axis direction, $\mathbf{a}$, and the radial base vector, $\mathbf{e}_r$ ($\varphi = 0$), can be specified as well. These direction vectors are automatically normalized.

The definitions of the cylindrical coordinates in terms of the global Cartesian coordinates $\mathbf{r} = \mathbf{r}(x, y, z)$ are

$$\begin{bmatrix} r \\ \varphi \\ a \end{bmatrix} = \begin{bmatrix} |\mathbf{r}-(\mathbf{r}_0+\mathbf{a}(\mathbf{a}\cdot(\mathbf{r}-\mathbf{r}_0)))| \\ \operatorname{atan}\dfrac{(\mathbf{a}\times\mathbf{e}_r)\cdot(\mathbf{r}-\mathbf{r}_0)}{\mathbf{e}_r\cdot(\mathbf{r}-\mathbf{r}_0)} \\ \mathbf{a}\cdot(\mathbf{r}-\mathbf{r}_0) \end{bmatrix}$$

To add a Cylindrical System to any Component:

- On the **Definitions** toolbar select **Cylindrical System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** ( ) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

Frame

Select with respect to which **Frame**—**Spatial** (the default), **Mesh**, **Material**, or **Geometry**—the coordinate system is cylindrical as defined by the above transformations. Note that the actual coordinate names—typically **(x, y, z)** or **(X, Y, Z)**—are displayed for each frame, indicating which frames actually differ from each other in the current model.

A coordinate system with **Frame** set to **Spatial** is orthonormal only in the spatial frame. Similarly, a **Material** system is orthonormal only in the material frame. Some physics require that coordinate systems used are orthonormal in a particular frame. For example, choose the **Material** frame if you want to use the coordinate system in a structural mechanics model.

Coordinate Names

In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `phi`, and `a`. In planar 2D models, `r` and `phi` are in-plane polar coordinates, and `a` is the out-of-plane coordinate. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Origin

Specify the location of the **Origin** of the cylindrical coordinate system in the global Cartesian system. The default is an origin coinciding with the one from the global system.

Longitudinal Axis

For 3D models, enter the **Longitudinal axis direction**. The default is the z direction in the global system.

Direction of Axis

For 3D models, specify the **Direction of axis** φ **=0**, where φ is the azimuthal angle. The default direction is the x direction in the global system.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

If you have the AC/DC Module and Particle Tracing Module, see Magnetic Lens: model library path **ACDC_Module/Particle_Tracing/magnetic_lens**.

Mapped System

Use a **Mapped System** (⊞) to create a system that defines a mapping from the frame coordinate system.

This method can deal with translated and rotated coordinate systems:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1(x, y, z) \\ u_2(x, y, z) \\ u_3(x, y, z) \end{bmatrix}$$

To add a Mapped System to any Component:

- On the **Definitions** toolbar select **Mapped System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

The **Frame** list is the same as for the Cylindrical System.

Coordinate Names

In the **Coordinate names** table, the default names are entered—`x1`, `x2`, and `x3`. In planar 2D models, `x1` and `x2` are typically the in-plane coordinates, and `x3` is the out-of-plane coordinate. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Coordinate Mapping

Under **Coordinate mapping**, the **Coordinate** column displays the **Coordinate names** with the **Expression** column displaying the associated mapped coordinate.

Simplifications

If required, select the **Assume orthonormal** check box. The program then uses the assumption that the settings define an orthonormal coordinate system.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

Rotated System

Use a **Rotated System** ( ) to define rotation about the out-of-plane axis in 2D and Euler angles in 3D.

In the **Rotated System** node's settings window, define the rotation relative to the global Cartesian coordinate system. In 3D models, you specify the local coordinate system (x_1, y_1, z_1) using three consecutive Euler angles (rotation angles) α, β, and γ. See Figure 5-2.

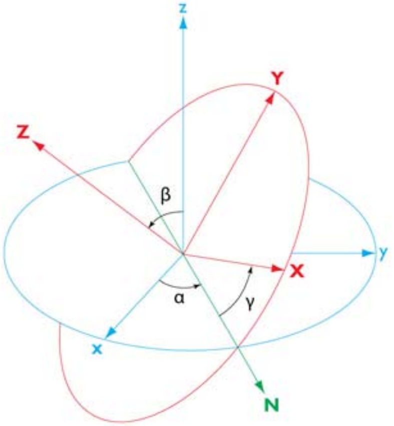


Figure 5-2: 3D Euler angles in a rotated coordinate system.

The transformation matrix for the 3D case is then

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \sin\beta\sin\alpha \\ \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\beta\cos\gamma & -\sin\beta\cos\alpha \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{bmatrix}^T \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

In 2D models, you describe the rotated coordinate system by the rotation angle about the out-of-plane vector. In both cases the origin of the coordinate system can be defined.

To add a Rotated System to any Component:

- On the **Definitions** toolbar select **Rotated System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

Coordinate Names

In the **Coordinate names** table, the default names are entered—x1, x2, and x3. In planar 2D models, x1 and x2 are typically the in-plane coordinates, and x3 is the out-of-plane coordinate. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Specify the location of the origin of the rotated coordinate system. Define it as a vector with two (for 2D) or three (for 3D) components. The default is the origin for the global Cartesian coordinate system. Using another origin translates the coordinates in the rotated system by that distance from the global Cartesian origin.

Out-of-plane Axis (2D)

For 2D models, select an out-of-plane axis from the **Out-of-plane axis** list (first, second, or third coordinate direction into or out-of screen), and then if necessary adjust the base vectors in the table under **Base vectors**. Enter the **Rotation about out-of-plane axis** (in radians). The default is 0.

Euler Angles (3D)

For 3D models, enter the **Euler angles (Z-X-Z)** (in radians) in the α, β, and γ fields (see the graphics in the settings window for definitions of these angles). The default values are 0 for all angles.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

If you have the MEMS Module, see Gecko Foot: model library path **MEMS_Module/Actuators/gecko_foot**.

Spherical System

Use a **Spherical System** ( ) to define a spherical coordinate system in 3D by its origin, zenith axis, and azimuth axis.

The coordinates of a local spherical coordinate system are (r, θ, φ), where r represents the radial distance from the origin, θ is the inclination (in the interval from 0 to π), and φ is the azimuthal angle (in the interval from −π to π). Specify—in terms of the global Cartesian coordinates x, y, and z—the position of the origin, the axis $\theta = 0$ (the zenith axis, **Z**), and the axis $\theta = \pi/2$, $\varphi = 0$ (the azimuth axis, **A**). The direction vectors are automatically normalized.

This is a mapped normalized coordinate system using the following transform in global coordinates

$$
\begin{aligned}
r &= |\mathbf{r}-\mathbf{r}_0| \\
\theta &= \mathrm{acos}\left(\frac{\mathbf{Z}\cdot(\mathbf{r}-\mathbf{r}_0)}{|\mathbf{r}-\mathbf{r}_0|}\right) \\
\varphi &= \mathrm{atan2}(\mathbf{r}_\perp\cdot(\mathbf{Z}\times\mathbf{A}), \mathbf{r}_\perp\cdot\mathbf{A})
\end{aligned}
$$

where $\mathbf{r}_0$ is the position of the origin, $\mathbf{Z}$ is a unit vector along the axis $\theta = 0$, and the component of $\mathbf{r}-\mathbf{r}_0$ in the plane $\theta = \pi/2$ is

$$\mathbf{r}_\perp = (\mathbf{r}-\mathbf{r}_0-\mathbf{Z}(\mathbf{Z}\cdot(\mathbf{r}-\mathbf{r}_0)))$$

To add a Spherical System to any Component:

- On the **Definitions** toolbar select **Spherical System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

The **Frame** list is the same as for the Cylindrical System.

In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `theta`, and `phi`. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Enter the location of the **Origin** in the global Cartesian coordinate system. The default is an origin coinciding with that of the global system.

Enter the **Direction of axis** θ**=0** (the *zenith axis*). The default axis direction is the z direction in the global Cartesian system.

Define the **Direction of axis** θ**=**π**/2,** φ **=0** (the *azimuth axis*). The default direction is the x direction in the global Cartesian system.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

Scaling System

Use a **Scaling System** ( ) to create a system that maps the geometry, as represented by the independent coordinates of an underlying frame, onto a virtual geometry represented by virtual scaling system coordinates. Physics interfaces that support infinite elements or perfectly matched layers accept the scaling system coordinates as being the physical domain, in which the underlying frame coordinates are seen as a parameterization. Therefore, using a scaling coordinate system you can arbitrarily deform the domain, essentially in the same way as when using Deformed Geometry with a Prescribed Deformation node.

The Scaling System is only available for physics that support infinite elements or perfectly matched layers. See Infinite Element Domain.

The scaling coordinate system is defined as a map from real frame coordinates to virtual scaling system coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1(x, y, z) \\ x_2(x, y, z) \\ x_3(x, y, z) \end{bmatrix}$$

The selected frame coordinates (the setting is invisible if there is only one frame) are seen as a parameterization of the “true geometry” in which the physics is solved. What you specify in the **Coordinate mapping** table is therefore a “true position” for each point in the mesh, expressed in the frame coordinates. When applied to a domain with a compatible material model in a physics interface, the equations in that domain are first reformulated in terms of the virtual x_1, x_2, and x_3 coordinates but then automatically mapped back to the frame coordinates. This leads to explicit transformation expressions appearing in the equations.

To add a Scaling System to any Component:

- On the **Definitions** toolbar select **Scaling System** from the **Coordinate Systems** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Coordinate Systems** submenu.

SETTINGS

Under **Coordinate mapping**, the **Coordinate** column displays the virtual coordinate names with the **Expression** column displaying the map from underlying frame coordinates to virtual coordinates. The default expressions are the spatial coordinates x, y, and z, which means no scaling.

Go to Coordinate System Identifier for information about the **Coordinate System Identifier** section.

Identity and Contact Pairs

Pairs are available for assemblies (that is, geometries created by not forming a union of all geometry objects as the final step), where there is a need to connect boundaries between parts. By default, pairs are created automatically when forming an assembly. There are two types of pairs—identity and contact.

About Identity and Contact Pairs

IDENTITY PAIRS

An *identity pair* ( ) is a pair that by default makes the fields across two connected boundaries (one from each connecting part in an assembly) continuous. This is equivalent to the continuity that is obtained by default on interior boundaries in a geometry created by forming a union. Some physics provide special boundary conditions for identity pairs to model "slit conditions" such as resistive layers. You can specify boundary conditions for these pairs from the **Pairs** submenu at the bottom of the boundary condition part of the context menu for the physics. The nodes in the **Model Builder** that represent pair boundary conditions use an icon with a pair symbol in the lower-left corner: .

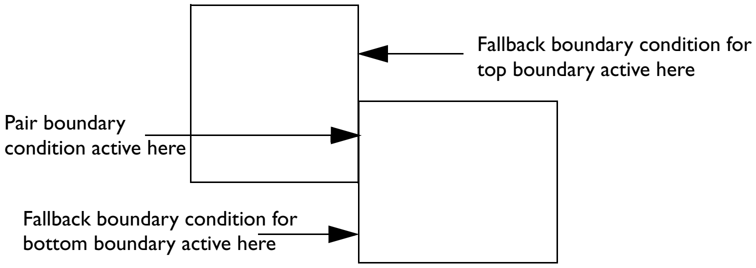
CONTACT PAIRS

A *contact pair* ( ) is a pair that define boundaries where the parts can come into contact but cannot penetrate each other under deformation for modeling of structural contact and multiphysics contact.

Contact pair modeling requires the Structural Mechanics Module or MEMS Module. Details about this pair type can be found in the respective user guide.

FALLBACK BOUNDARY CONDITIONS ON NON-OVERLAPPING PARTS

For pairs where parts of the boundaries do not overlap you need to specify boundary conditions for the non-overlapping parts, which typically represent exterior boundaries outside of the overlapping area. These boundary conditions (*fallback boundary conditions*) appear as subnodes to the pair's boundary condition node in the **Model Builder**. By default, the default boundary condition for exterior boundaries is added to the non-overlapping parts. If you want to use another boundary condition for any of the non-overlapping parts, right-click the pair's boundary condition node (**Continuity**, for example) and select any of the standard boundary conditions from the **Fallback Features** submenu. In the settings window, the selection includes all applicable boundaries by default, but a separate boundary condition can be added for only a subset of the pair boundaries. In the following illustration, which shows a simple example with two partially overlapping rectangles, there is one identity pair that consists of two boundaries, each with a non-overlapping part. You can right-click the pair's boundary condition node and, from the **Fallback Features** submenu, add one fallback boundary condition for the top boundary and another fallback boundary condition for the bottom boundary if desired.


Fallback boundary condition for top boundary active here
Pair boundary condition active here
Fallback boundary condition for bottom boundary active here

The options for the available fallback conditions are based on the physics interface and the license type (see Figure 5-3).

With only a few exceptions for the Solid Mechanics interface or other physics interfaces using Solid Mechanics functionality, all subnodes to pairs are fallback nodes.

When additional fallback feature nodes are added, the node has an indicator in the lower-left corner ( ) identifying it as a fallback feature node.

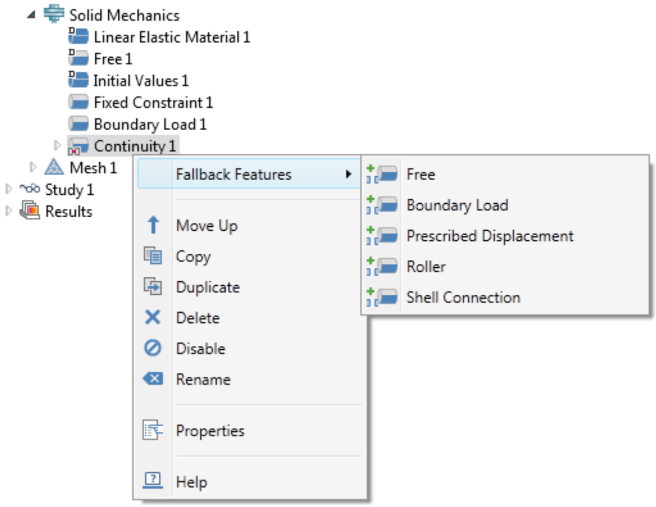


Figure 5-3: An example of the Fallback Features submenu for a Continuity pair added to the Solid Mechanics interface.

To add Pairs to any Component:

- On the **Definitions** toolbar select features from the **Pairs** menu, or
- Right-click the **Definitions** (≡) node and choose an option from the **Pairs** submenu.

Identity Pair

Use an **Identity Pair** node (for an identity boundary pair) to specify two selections of boundaries that overlap but belong to different parts of an assembly. Then assign a boundary condition to connect the physics in the two parts in a physics interface. Identity pairs connect overlapping boundaries in different connecting parts of an assembly.

The **Identity Boundary Pair** ( ) is the most commonly used. For 3D models, the **Identity Edge Pair** is available (, which can be useful for connecting two edges in a shell model, for example. You can also choose the **Identity Point Pair** ( ).

To add an Identity Pair to any Component:

- On the **Definitions** toolbar select **Identity Boundary Pair**, **Identity Edge Pair**, or **Identity Point Pair** from the **Pairs** menu, or
- Right-click the **Definitions** (≡) node and choose an option from the **Pairs** submenu.

GENERAL

Enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair. The default **Pair type** is **Identity pair**, or select **Contact pair**. Identity and Contact pairs are both available for boundary pairs, and Identity pairs are also available on edge and point levels.

Manual Control of Selections

If the pair was created automatically when forming an assembly, the **Manual control of selections** check box is visible. Click to clear this check box to be able to make manual changes to the **Source Boundaries** and **Destination Boundaries** selections. Pairs in manual mode do not have their selections updated when the geometry sequence is rebuilt.

The operator mapping an expression E on the source side to the destination side is denoted `src2dst_pn(E)`, where `_pn` is the pair name.

For an **Identity Pair**, the variable `src2dst_pn` (defined on the destination) is `1` where there is a corresponding source point, and `0` otherwise. The corresponding operator and variable for use on the source side are denoted `dst2src_pn`.

Similarly, for a **Contact Pair** there is an operator `src2dst_pn_mph` that is suited for use in multiphysics coupling. The variable `geomgap_dst_pn` is the geometric gap between the source and the destination, seen from the destination side (following the normal of the destination boundary). The corresponding operators and variables for use on the source side are denoted `dst2src_pn`, `dst2src_pn_mph`, `geomgap_src_pn`.

SOURCE BOUNDARIES AND DESTINATION BOUNDARIES

The destination boundaries should overlap the source boundaries. The condition that connects the physics on the destination and source boundaries is specified in the physics interface. For example, it can be a constraint that constrains a dependent variable (temperature, for example) on the destination side to be equal to a dependent variable on the source side.

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination boundaries. Select **Manual** or **All boundaries** for the boundaries on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section. If required, click the **Swap Source and Destination** button (↕).

Select **Manual** or **All boundaries** for the boundaries on the source side or the destination side. If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section. If required, click the **Swap Source and Destination** button (↕) to swap the source boundaries and the destination boundaries.

For **Identity Edge Pairs** and **Identity Point Pairs**, edges and points, respectively, replace boundaries in the selections of the pair's source and destination.

FRAME

If there are several frames in the model, the **Frame** section is visible. Select the **Source frame** and the **Destination frame**. Source and destination points are connected if their coordinates in their respective frames are equal.

Thin-Layer Diffusion: model library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**

Contact Pair

Use a **Contact Pair** node ( ) to specify two selections of boundaries that cannot penetrate each other under deformation. The contact pairs define boundaries for parts that can come into contact (boundaries that cannot penetrate each other under deformation). For more information about contact modeling and guidelines for selecting source and destination boundaries for contact pairs, see the Structural Mechanics Module or MEMS Module documentation.

To add a Contact Pair to any Component:

- On the **Definitions** toolbar select **Contact Pair** from the **Pairs** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Pairs** submenu.

GENERAL

This section is the described for the Identity Pair except that the default **Pair type** is **Contact pair**.

SOURCE BOUNDARIES AND DESTINATION BOUNDARIES

The contact algorithm constrains the destination boundaries so that they do not penetrate the source boundaries.

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

Then define the source or destination boundaries. Select **Manual** or **All boundaries** for the boundaries on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section. If required, click the **Swap Source and Destination** button (↕) to swap the source boundaries and the destination boundaries.

ADVANCED

The **Search method** defaults to **Fast**—the algorithm only keeps track of source and destination points that have a distance less than a certain *search distance*. Select **Direct** for a slower but more robust search.

Select the **Manual control of search distance** check box to tune the search distance (SI unit: m). By default, the search distance is taken as 0.01 times the diagonal of the geometry's bounding box. If the **Manual control of search distance** check box is selected, enter a different value in the **Distance** field.

For a contact pair, the fallback boundary condition is applied to all parts of the boundaries currently not in contact.

- If you have the MEMS Module, see Microgripper: model library path **MEMS_Module/Piezoelectric_Devices/microgripper**.
- If you have the Nonlinear Structural Materials Module, see Snap Hook: model library path **Nonlinear_Structural_Materials_Module/Plasticity/snap_hook**.
- If you have the Structural Mechanics Module, see Cylinder Roller Contact: model library path **Structural_Mechanics_Module/Verification_Models/cylinder_roller_contact**.

Probes

About Probes

Probes ( ) monitor the development of a scalar-valued quantity (real or complex-valued number) from a time-dependent, frequency-domain, or parametric simulation by two different results presentations: tabulated data and 1D graph plots. You can probe while solving, as a monitor and diagnostic tool, and probe after the computation is finished for results analysis. On top of this functionality, a probe variable in the model component's namespace and with a global evaluation scope is also defined. The probe variable's name appears in the **Probe variable** field. You can use this variable can be used as any other variable in, for example, equations, boundary conditions, or a stop condition.

Plot while solving is a technique used to briefly interrupt the simulation and launch some predefined plot commands and then continue with the simulation. Both normal plots and graphs can be plotted for probes during the simulation.

There are these types of probes (see Table 5-16 for the icon by space dimension):

- *Domain probes*, *boundary probes*, and *edge probes* make it possible to probe the average, minimum, maximum, or integral of a field quantity over a domain, on a boundary, or along an edge.
- *Domain point probes* and *boundary point probes* provide the value of some field quantity at a point in the domain or on a boundary. Any point within the domain or on the boundary can be defined.
- Use *Global variable probes* ( ) for probing the value of any global variable.

The probes automatically create a **Probe Table** node for displaying numerical results in the **Table** window and an associated plot group with a **Probe Table Plot** node that plots the probe data as a line graph in a separate **Probe Plot** window. For further processing, the probes also add data sets such as **Domain Point Probe** data sets ( ), which give access to the probe data. For further control, specify the table and plot window each probe uses.

To add a Probe to any Component:

- On the **Definitions** toolbar select features from the **Probes** menu, or
- Right-click the **Definitions** ( ) node and choose an option from the **Probes** submenu.

When the simulation has finished, click the **Update Results** button ( ) in the probe settings window (or on the **Definitions** toolbar) to change the settings for a probe and update the results information. Then right-click the **Definitions** node and select **Update Probes**.

TABLE 5-16: PROBE TYPES AND ICONS BY SPACE DIMENSION

PROBE TYPE	3D	2D	1D
Domain			
Boundary			
Edge		—	—
Domain Point			—

TABLE 5-16: PROBE TYPES AND ICONS BY SPACE DIMENSION

PROBE TYPE	3D	2D	1D
Boundary Point	🌡	—	—
Global Variable	🌡	🌡	🌡

Getting Results While Solving

Common Settings for Probes

SOURCE SELECTION

The source selection defines the source for the probes—the part of the geometry over which the program computes the probes.

From the **Geometric entity level** list, select **Manual** or based on the probe type, **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list. If **Manual** is selected, select geometric entities in the **Graphics** window.

EXPRESSION

In the **Expression** section you can:

- Enter a text string with your own expression.
- Click the **Replace Expression** () button to select a predefined quantity and replace the contents of the **Expression** field with the corresponding variable.
- Click the **Insert Expression** () button to insert the corresponding variable at the current position in the **Expression** field.
- Select a **Table and plot unit** from the list. You can select from a predefined number of applicable units for the quantity that the variable represents, but you can also click in the unit's text field and type any compatible unit for that quantity to use a unit that is not in the list (for example, `mi/h` for miles per hour as a unit for a velocity quantity).
- Select the **Description** check box to enter a description (or edit the default).

- About Parameters, Variables, and Expressions
- Entering Ranges and Vector-Valued Expressions
- Expressions and Predefined Quantities

TABLE AND WINDOW SETTINGS

By default, COMSOL uses a probe table (typically **Probe Table 1**) under **Tables** and a probe table plot (typically **Probe Table Plot 1**) in a **Probe 1D Plot Group** node, which appears in a separate plot window for probe plots (typically **Probe Plot 1**). To organize and group multiple probes, control the table and plot window to use for the probe results:

From the **Output table** list, select **Default**, **New table**, or any existing probe table. If an existing probe table is selected, click the **Go to Source** button (⇥) to move to the selected **Probe Table** node under **Tables**.

From the **Plot window** list, select **Default**, **New window**, or any existing plot window. Click the **Add Plot Window** button (➕) to create a new plot window and make it the default for this list.

By default for both the **Output table** and **Plot window**, COMSOL uses a probe table or probe table plot that is created automatically. If **Default** is selected, COMSOL updates the list to show the name of the default probe table or probe plot window after the solution process.

Domain Probe, Boundary Probe, and Edge Probe

Use a **Domain Probe** ( ), **Boundary Probe** ( ), or **Edge Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency-domain, or parametric solution).

> Go to Common Settings for Probes for information about the **Source Selection**, **Expression**, and **Table and Window Settings** sections.

To add a Probe to any Component:

- On the **Definitions** toolbar select **Domain Probe**, **Boundary Probe**, or **Edge Probe** from the **Probes** menu, or
- Right-click the **Definitions** (≡) node and choose an option from the **Probes** submenu.

PROBE SETTINGS

Select an option from the **Type** list—**Average** (the default), **Maximum**, **Minimum**, or **Integral** depending on what type of value takes over the domain, boundary, or edge that you want the probe to compute and output. If needed, enter or edit a name for the **Probe variable**. The defaults are `dom1` for a Domain Probe, `bnd1` for a Boundary Probe, and `edge1` for an Edge Probe.

INTEGRATION SETTINGS

If you have selected **Average** or **Integral** from the **Type** list, the **Integration Settings** section contains the following settings:

- From the **Method** list, select **Integration** (the default) or **Summation**. Only reaction forces use the summation method.
- Enter an integer value in the **Integration order** field (default: 4).

If you have selected **Maximum** or **Minimum** from the **Type** list, the **Integration Settings** section contains an **Element refinement** field, where you can enter the element refinement (number of partitions of an element edge) to control the accuracy of the maximum or minimum value (default value: 4).

When working with multiple frames for any type of probe, you can also select a **Frame**—**Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics, for the volume element to be used in the integration.

- If you have the Batteries & Fuel Cells Module: For a boundary probe example, see Edge Effects in a Spirally Wound Li-Ion Battery: model library path **Batteries_and_Fuel_Cells_Module/Batteries/li_battery_spiral_2d**. For a domain probe example, see Mass Transport Analysis of a High Temperature PEM Fuel Cell: model library path **Batteries_and_Fuel_Cells_Module/PEMFC/ht_pem**.
- If you have the Nonlinear Structural Materials Module and for a boundary probe example, see Snap Hook: model library path **Nonlinear_Structural_Materials_Module/Plasticity/snap_hook**.

Domain Point Probe

Use a **Domain Point Probe** ( ) to monitor the development of a real or complex-valued number from a dynamic simulation (a time-dependent, frequency-domain, or parametric study). By default a Point Probe Expression subnode is added, or right-click **Domain Point Probe** to add additional nodes.

To add a Domain Point Probe to any Component:

- On the **Definitions** toolbar select **Domain Point Probe** from the **Probes** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Probes** submenu.

POINT SELECTION

When working with multiple frames, select a **Frame**—**Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics.

For 3D models, select a **Line entry method**—**Point and surface normal** (the default), **Point and direction**, **Two points**, or **None**.

For all models, enter **Coordinates**—Enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D). Also select the **Snap to closest boundary** check box to snap the selected points to the grid.

For **Point and surface normal**, **Point and direction**, or **Two points**, enter a **Depth along line** or use the slider to select a value between 0 and 1 to determine the probe location along the line anywhere from the starting point (0) to the ending point (1).

- For **Point and surface normal** click at a position on the surface of the geometry. The direction becomes the inward surface normal as defined by the geometry, which for an exterior boundary means that the probe location can be anywhere from the start position to the end of the geometry in the normal direction.
- For **Point and direction**, the direction becomes that of a ray directed away from the point in the current camera view (that is, the direction depends on the view).

For **Two points**, from the **Point being modified** list, also select **First point** and click on the geometry to define the first point (starting point). Then select **Second point** and click to define the second point (ending point) for the line.

Process Control Using a PID Controller: model library path **COMSOL_Multiphysics/Multidisciplinary_Models/pid_control**

Boundary Point Probe

Use a **Boundary Point Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency, parametric). By default a Point Probe Expression subnode is added, or right-click **Boundary Point Probe** to add additional subnodes.

To add a Boundary Point Probe to any Component:

- On the **Definitions** toolbar select **Boundary Point Probe** from the **Probes** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Probes** submenu.

BOUNDARY SELECTION

Select a single boundary to add to the **Selection**.

POINT SELECTION

When working with multiple frames, select a **Frame**—**Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics.

Enter the **Coordinates**. A red dot indicates the position of the point on the selected surface in the **Graphics** window. Click the surface to move the point, or enter **x**, **y**, and **z** coordinates. If the point is not on the boundary, the probe location becomes the closest point on the boundary, with coordinates indicated by **On surface** under the fields.

Point Probe Expression

A **Point Probe Expression** ( ) is automatically added as a subnode to a Domain Point Probe and a Boundary Point Probe. Right-click the main node to add additional **Point Probe Expression** subnodes. Under **Probe Settings**, edit or enter a name for the **Probe variable**. The default name is `ppb1`.

To add a Point Probe Expression to any Component:

- On the **Definitions** toolbar select **Point Probe Expression** from the **Probes** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Probes** submenu.

Go to Common Settings for Probes for information about the **Expression**, and **Table and Window Settings** sections.

Global Variable Probe

Use a **Global Variable Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex-valued number) from a dynamic simulation (time-dependent, frequency, parametric). Under **Probe Settings**, edit or enter a name for the **Probe variable**. The default name is `var1`.

To add a Global Variable Probe to any Component:

- On the **Definitions** toolbar select **Global Variable Probe** from the **Probes** menu, or
- Right-click the **Definitions** (≡) node and choose it from the **Probes** submenu.

Go to Common Settings for Probes for information about the **Expression**, and **Table and Window Settings** sections.

If you have the Plasma Module or MEMS Module:

- See Harmonic Content of the Power Deposition into a Dual Frequency Capacitively Coupled Plasma: model library path **Plasma_Module/Capacitively_Coupled_Plasmas/harmonic_content**.
- See Transient Response of a Biased Resonator—2D: model library path **MEMS_Module/Actuators/biased_resonator_2d_transient**.

Infinite Element Domains and Perfectly Matched Layers

Simulation of Infinite Domains

Simulation of unbounded or infinite domains is a challenge encountered in many types of physics. Normally, any physics simulates a process within a bounded domain represented by the geometry drawn in, or imported into, COMSOL. But the domain is often delimited by artificial boundaries inserted to limit the extent of the model to a manageable *region of interest*. You might not be interested in the details of the solution far away from any sources or material inhomogeneities, but the solution inside the region of interest must not be affected by the presence of the artificial boundaries. You simply want it to behave as if the domain was of infinite extent.

Artificial truncation of the domain can be handled in several ways. Some physics interfaces include special boundary conditions to absorb outgoing propagating waves without spurious reflections, so-called low-reflecting boundary conditions. Others allow impedance boundary conditions, which can account for a finite impedance between the model boundary and a reference at infinity. Such boundary conditions are often efficient and useful but lack some generality and sometimes accuracy.

Another way to accomplish the same desired effect is to apply a coordinate scaling to a layer of virtual domains surrounding the physical region of interest. For stationary and transient problems, these virtual domains can be stretched out toward infinity, giving rise to *infinite elements*. To absorb outgoing waves in a frequency-domain problem, you must instead stretch the virtual domains into the complex plane, creating so-called *perfectly matched layers* (PMLs).

Because of their common background as coordinate stretching, infinite elements and PMLs in COMSOL share a number of important properties.They share part of the user interface and many modeling principles can be translated directly from one to the other. In the description below, infinite elements and PMLs are therefore sometimes referred to collectively as *scaling systems*.

The Scaling System node provides direct access to coordinate transformation machinery underlying PMLs and infinite elements.

Standard Geometry Configurations

Automatic scaling systems are available in COMSOL for three distinct geometrical configurations: **Cartesian**, **Cylindrical**, and **Spherical**. Which ones you can use depends on the space dimension of the Component.

Plane 2D Models

The available scaling types in plane 2D models are **Cartesian** and **Cylindrical**. Cartesian domains are stretched in one or two directions depending on whether they are attached to an edge or to a corner of the physical region of interest.

It is important that separate, normally quadratic, domains are drawn at the corners.

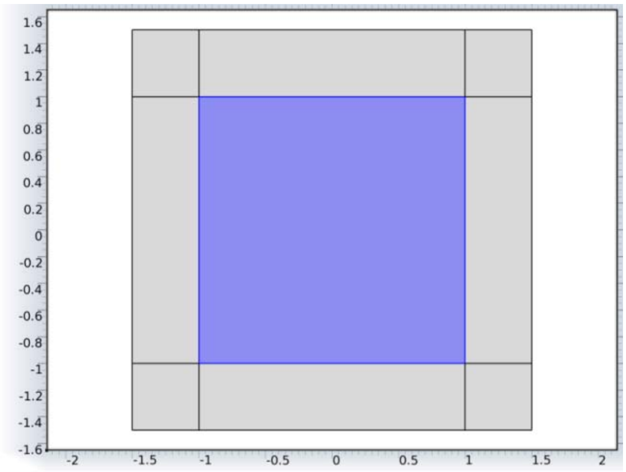


Figure 5-4: Typical Cartesian scaling configuration. Note the distinction between edge and corner domains.

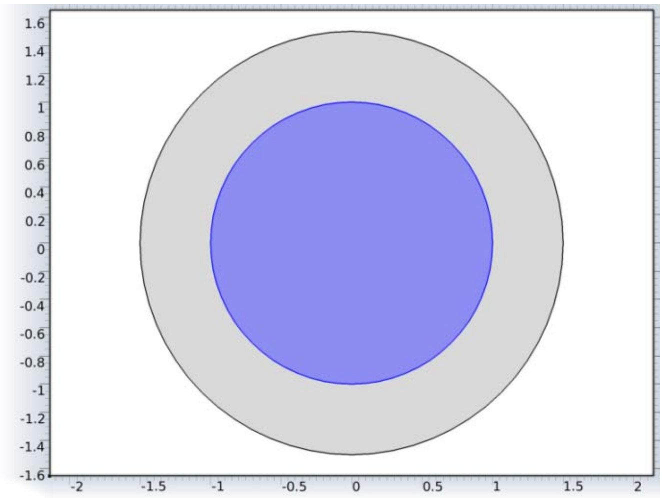


Figure 5-5: Example of cylindrical scaling configuration in plane 2D. You must specify the center point of the model when different from the origin of the coordinate system.

AXISYMMETRIC 2D MODELS

The available scaling types in 2D axisymmetric models are **Cylindrical** and **Spherical**. The axisymmetric cylindrical configuration, from the practical point of view, behaves identically to the plane 2D Cartesian option. Similarly, the axisymmetric spherical scaling is similar to plane 2D cylindrical scaling, except that it is always centered on the axis.

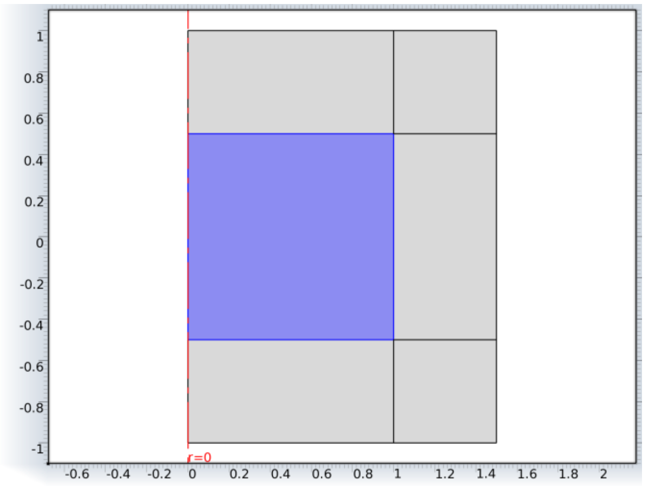


Figure 5-6: Axisymmetric cylindrical scaling uses domains of three distinct types: with radial stretching, with axial stretching, and with both radial and axial stretching. The latter are the corner zones, which must be drawn as distinct domains.

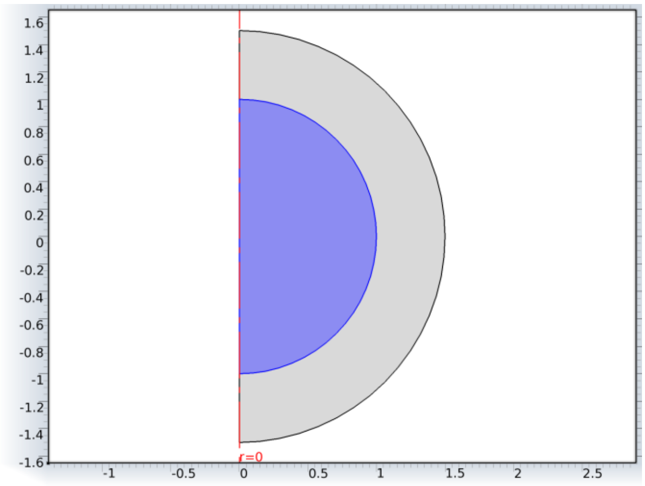


Figure 5-7: Axisymmetric spherical scaling assumes radial stretching in an annulus centered at a point on the axis. If the centerpoint is not the origin of the coordinate system, you must specify its axial position.

3D MODELS

The available scaling types in 3D are **Cartesian**, **Cylindrical** and **Spherical**. The Cartesian scaling domains are of three different types. Depending on whether they are attached to a surface, an edge, or a point in the physical domain, they are stretched in one, two, or three directions, respectively. Cylindrical scaling domains are also of three different types: radially stretched, axially stretched, and stretched both radially and axially. Spherically scaled domains are always stretched only in the spherical domain's radial direction.

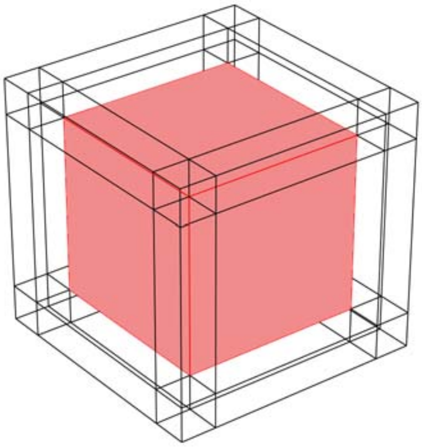


Figure 5-8: There are three different types of Cartesian scaling domain, attached to faces, edges and corners, respectively. They differ in the number of scaled directions. Note that the edge and corner zones must be drawn as distinct domains in the geometry.

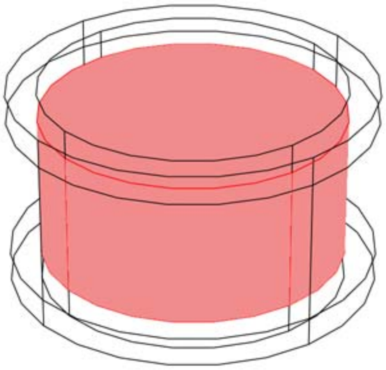


Figure 5-9: The three different types of cylindrical scaling domain are attached to the sides, top and bottom, and edges of the cylindrical physical domain. The position and orientation is specified as a center point and an axial direction. The scaling system domains are stretched in the radial direction, away from the axis, in the axial direction, and in both radial and axial direction, respectively.

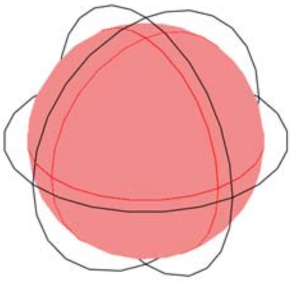


Figure 5-10: A spherical scaling domain stretches the coordinate system only in the radial direction relative to a specified center point.

Note on Availability

Infinite elements and perfectly matched layers are available only for some physics and when COMSOL is used together with certain add-on modules. If you have not added any physics that is compatible with infinite elements or perfectly matched layers under the available licenses, you cannot add such features to the model.

Further, after turning a domain into a PML or infinite element, that domain is not allowed in the active selection of physics interfaces and individual nodes that are not compatible with these special domain types. This means that the scaled domains are either not selectable at all or display as **Not applicable** in the selection list.

PML Implementation

PMLs apply a complex coordinate stretching in one, two, or three directions, depending on how the PML domain connects to the physical domain. In each direction, the same form of stretching is used, defined as a function of a dimensionless coordinate ξ, which varies linearly from 0 to 1 over the PML layer. For a PML absorbing waves propagating in the direction of unit vector $\mathbf{n}_\xi$, ξ is defined as

$$\xi = \frac{\mathbf{n}_\xi \cdot \mathbf{x} - x_0}{\Delta_w}$$

where **x** is the original, unscaled, coordinate vector, Δ_w is the original width of the PML (as drawn in the geometry), and x_0 is a reference distance defined such that $\xi = 0$ whenever **x** lies on the boundary between the PML and the physical domain. For a cylindrical or spherical PML, for example, x_0 is its radius.

The corresponding scaled coordinate vector **x'**, in which the equations are reformulated inside the PML, is defined as

$$\mathbf{x}' = \mathbf{x}_0 + \mathbf{n}_\xi f(\xi)$$

where $f(\xi)$ is a scalar stretching function, and $\mathbf{x}_0$ is the projection of the unscaled coordinate vector **x** onto the surface $\xi = 0$:

$$\mathbf{x}_0 = \mathbf{x} - \mathbf{n}_\xi(\mathbf{n}_\xi \cdot \mathbf{x}) + x_0$$

In the PML nodes, you can choose between a *polynomial* stretching function and a *rational* stretching function. The polynomial stretching function is defined as

$$f_p(\xi) = s\lambda\xi^p(i-1)$$

where λ is a *typical wavelength* parameter, p is a *curvature* parameter, and s is a *scaling factor*. The rational stretching function is defined as

$$f_r(\xi) = s\lambda\xi\left(\frac{1}{3p(1-\xi)+4} - \frac{i}{3p(1-\xi)}\right)$$

The quantities $\mathbf{n}_\xi$, x_0, and Δ_w are computed automatically based on the selected geometry type and an analysis of the actual geometry. The typical wavelength λ is normally supplied by a physics interface while p and s are user inputs.

> There is no check that the geometry of the region is correct, so it is important to draw a proper geometry and select the corresponding region type.

INTERPRETING PML PARAMETERS

The PML coordinate stretching is controlled by three parameters:

- The *typical wavelength* represents the longest wavelength of propagating waves in an infinite medium. It is normally provided by a physics interface. For nondispersive media, it is expected to be inversely proportional to the frequency and serve to make the PML perform similarly for all frequencies.

> In eigenfrequency studies, the typical wavelength parameter must not depend on the—unknown—frequency. When the typical wavelength is set to be obtained from a physics interface, it is therefore redefined to be equal to the PML width Δ_w instead. A user-defined typical wavelength applies as entered, but must not be a function of the frequency. It is often most convenient to draw and mesh the PML as if it had been part of the physical domain. To tune its effective thickness, use the scaling factor.

- The PML *scaling factor* multiplies the *typical wavelength* to produce an effective scaled width for the PML. For example, to retain perfect absorption for plane waves incident at an angle θ relative to the boundary normal, it is necessary to compensate for the longer wavelength seen by the PML in the stretching direction. In this case, $1/\cos(\theta)$ is a suitable scaling factor.

  Conversely, if resolving the field inside the PML proves too costly, it is possible to lower the scaling factor below its default value 1, to make better use of the available mesh elements. Note that this has a price in terms of less efficient absorption.
- The PML *curvature* parameter serves to relocate mesh resolution inside the PML. When there are components present which decay inside the PML much faster than the longest waves, the resolution must be increased in the zone closest to the boundary between PML and physical domain. Increasing the curvature parameter effectively moves available mesh elements toward the inner PML boundary. This is often necessary when the wave field contains a mix of different wavelengths or a mix between propagating and evanescent components.

> If you increase the curvature factor, you must normally still resolve the long propagating waves sufficiently, so an overall increase of the number of mesh elements across the PML is called for.

CHOOSING A STRETCHING TYPE

Which coordinate stretching type is most appropriate depends on the problem at hand. Consider the following when choosing between polynomial and rational stretching:

Polynomial The polynomial stretching strategy makes a minimum of assumptions about the wave field incident on the PML. Its finite and equal real and imaginary parts mean that propagating and evanescent waves with the same length scale are treated alike. The default scaling factor gives a PML with a maximum attenuation of about 109 dB for normal incidence and provided sufficient mesh resolution.

The polynomial stretching is generally applicable and most appropriate when there is a mix of different wave types in the model and you can afford at least 8 mesh elements across the PML. Also, compared to the rational stretching, it interferes less with the convergence of iterative linear solvers.

Rational The rational stretching is designed for propagating waves of mixed wavelengths and angles of incidence. The real part of the stretching scales the effective PML thickness to a quarter of a typical wavelength, while the imaginary part—responsible for the attenuation—is stretched out toward infinity. This means that provided sufficient mesh resolution, the PML absorbs any propagating wave perfectly.

In reality, the mesh resolution limits the effectiveness of the rationally stretched PMLs. For a single wavelength at normal incidence, 3 mesh elements across the PML normally give sufficient attenuation and accuracy. If the wave field contains also longer- or shorter-wavelength components, the mesh resolution must be increased. When other wave components are shorter than the supplied typical wavelength, increasing the curvature factor may be useful to make best use of the available resolution.

PMLs in Multiphysics

The coordinate stretching used in the PMLs is by default controlled by one of the physics interfaces in the model, which provides a *typical wavelength*. If each PML region contains a single active physics, and the PML regions are disjoint, you can set up separate PML nodes and choose different physics interfaces as wavelength source. If, instead, there are multiple physics active in the same PML domains or in adjacent domains—such as when an air-water interface extends into the PML—you must choose a single typical wavelength. Either choose a wavelength provided by one of the interfaces, or set a user-defined wavelength.

The way the stretching functions are defined, it usually makes the most sense to select the longest wavelength of propagating waves actually excited and propagating into the PML. Any shorter wavelengths must be accounted for by increasing the mesh resolution and curvature factor in the PML.

In the Acoustics Module, MEMS Module, and Structural Mechanics Module, you can control the typical wavelength passed from the physics interface to the PML, by changing the **Typical Wave Speed** property in the physics interface's settings window. The default wave speed generally corresponds to a compressional or pressure wave, which is the fastest wave type and therefore of longest wavelength. In the RF Module, the default for the typical wavelength is $2\pi/k$, where k is the local wavenumber.

Perfectly Matched Layer

A **Perfectly Matched Layer** node ( ) applies a complex coordinate scaling to a layer of virtual domains surrounding the physical region of interest. When appropriately tuned, this layer absorbs all outgoing wave energy in frequency-domain problems, without any impedance mismatch—causing spurious reflections—at the boundary.

To add a Perfectly Matched Layer to any Component, on the **Definitions** toolbar click **Perfectly Matched Layer**.

COORDINATE SYSTEM IDENTIFIER

The default **Identifier** is `pml1`. The identifier provides a namespace for variables created by the **Perfectly Matched Layer** node. For example, the scaled *x* coordinate can typically be accessed in equations and postprocessing as `pml1.x`. See the **Equation View** subnode for a complete list of available variables.

To display the **Equation View** node under all nodes creating variables, click the **Show** button and select **Equation View**. See also Equation View.

DOMAIN SELECTION

Select a set of domains conforming to the selected geometry type. See Standard Geometry Configurations.

GEOMETRY

Select a **Type**—**Cartesian** (the default), **Spherical**, or **Cylindrical**.

- If **Spherical** is selected, enter the position of the center of the spherical geometry in the **Center coordinate** table. For axisymmetric models, only the *z* coordinate is required since the geometry must be centered on the axis.
- If **Cylindrical** is selected, enter the position of a point on the cylinder axis in the **Center coordinate** table. For 3D models, also enter a **Center axis direction** vector.

SCALING

Select a **Coordinate stretching type**—**Polynomial** (the default) or **Rational**. See PML Implementation for help on making a decision.

Select an option from the **Typical wavelength from** list—**Physics interface** (the default) or **User defined**. If **Physics interface** is selected, select one of the interfaces supporting PMLs from the **Physics** list. If **User defined** is selected, enter a value or expression for the **Typical wavelength**. The default is 1.

Note that the **Physics interface** setting has no effect in Eigenfrequency studies. In that case, the typical wavelength is redefined to be equal to the PML width, as drawn in the geometry. The **User defined** option applies unaltered.

Enter a value or expression for the **PML scaling factor** and the **PML scaling curvature parameter** which can be used to tune the PMLs for wave fields with evanescent components or wavelengths deviating from the free-space wavelength of plane waves. See further PML Implementation. The defaults are 1 for both.

If you have the Acoustics Module, see the models:

- Cylindrical Subwoofer: model library path **Acoustics_Module/Tutorial_Models/cylindrical_subwoofer**
- Acoustic Scattering off an Ellipsoid: model library path **Acoustics_Module/Tutorial_Models/acoustic_scattering**

If you have the RF Module, see the models:

- Tutorial Models/Radar Cross Section (2D, cylindrical PML).
- Tutorial Models/RF Coil (3D, spherical PML with swept mesh).

Known Issues When Modeling Using PMLs

When modeling with PMLs be aware of the following:

USE OF ONE SINGLE PERFECTLY MATCHED LAYER NODE

A separate Perfectly Matched Layer node must be used for each simply connected PML region. That is, to use one and the same Perfectly Matched Layer node, all PML domains must be in contact with each other. Otherwise the PMLs do not work properly.

ELEMENT QUALITY

The coordinate scaling resulting from PMLs also yields an equivalent scaling of the mesh that can effectively result in a poor element quality. (The element quality displayed by the mesh statistics does not account for this effect.) This typically happens when the geometrical thickness of the PML deviates much from one wavelength (local wavelength rather than free space wavelength). The poor element quality causes poor convergence for iterative solvers and makes the problem ill-conditioned in general.

For this reason, it is strongly recommended that you use swept meshing in the PML domains. The *vector element* formulations (the ones using two or more components of a vector field variable) in the RF Module are particularly sensitive to low element quality. The sweep direction should be selected the same as the direction of scaling. For Cartesian PMLs and regions with more than one direction of scaling it is recommended to first sweep the mesh in the domains with only one direction of scaling, then sweep the domains with scaling in two directions, and finish by sweeping the mesh in the domains with PML scaling in all three directions.

COMPLICATED EXPRESSIONS

The expressions resulting from the stretching get quite complicated for spherical PMLs. This increases the time for the assembly stage in the solution process. After the assembly, the computation time and memory consumption is comparable to a problem without PMLs. The number of iterations for iterative solvers might increase if the PML regions have a coarse mesh.

ERRONEOUS RESULTS

PML regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the PML parameter to give erroneous results.

USE THE SAME MATERIAL PARAMETERS OR BOUNDARY CONDITIONS

The PML region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an PML region.

Infinite Element Implementation

Infinite elements apply a semi-infinite coordinate stretching in one, two, or three directions, depending on how the infinite element domain connects to the physical domain. In each direction, the same form of stretching is used, defined as a function of a dimensionless coordinate ξ, which varies linearly from 0 to 1 over the infinite element layer. For an infinite element domain stretched in the direction of unit vector $\mathbf{n}_\xi$, ξ is defined as

$$\xi = \frac{\mathbf{n}_\xi \cdot \mathbf{x} - x_0}{\Delta_w}$$

where $\mathbf{x}$ is the original, unscaled, coordinate vector, Δ_w is the original thickness of the infinite element domain (as drawn in the geometry), and x_0 is a reference distance defined such that $\xi = 0$ whenever $\mathbf{x}$ lies on the boundary between the infinite element and the physical domain. For a cylindrical or spherical infinite elements, for example, x_0 is its radius.

The corresponding scaled coordinate vector **x'**, in which the equations are reformulated inside the infinite element domains, is defined as

$$\mathbf{x}' = \mathbf{x}_0 + \mathbf{n}_\xi f(\xi)$$

where $f(\xi)$ is a scalar stretching function, and $\mathbf{x}_0$ is the projection of the unscaled coordinate vector $\mathbf{x}$ onto the surface $\xi = 0$:

$$\mathbf{x}_0 = \mathbf{x} - \mathbf{n}_\xi(\mathbf{n}_\xi \cdot \mathbf{x}) + x_0$$

The stretching function is defined as

$$f(\xi) = \frac{\xi}{\gamma - \xi}\Delta_p \tag{5-1}$$

where Δ_p is the, so called, pole distance and γ is a number larger than one, computed as

$$\gamma = \frac{\Delta_s + \Delta_p}{\Delta_s}$$

where Δ_s is the scaled thickness of the infinite element domain.

The quantities $\mathbf{n}_\xi$, x_0, and Δ_w are computed automatically based on the selected geometry type and an analysis of the actual geometry. The scaled thickness Δ_s and the pole distance Δ_p are user inputs.

!

There is no check that the geometry of the region is correct, so it is important to draw a proper geometry and select the corresponding region type.

INTERPRETING INFINITE ELEMENT PARAMETERS

The infinite element stretching has two user-defined parameters which let you control the thickness of the quasi-infinite region, as perceived by the physics interfaces, as well as how the stretching is distributed across the domain.

Physical width The scaled width of the infinite element domain, Δ_s, is by default set to `1e3*dGeomChar`, where the constant `dGeomChar` is a characteristic geometry dimension. The domain is therefore by default scaled to be very much larger than the original geometry, but not quite infinite in order to avoid numerical difficulties. In particular, the finite distance to the far-away boundary allows prescribing standard boundary conditions effectively at infinity.

Pole distance The coordinate stretching function, Equation 5-1, used in the infinite element domain contains a singularity when $\xi = \gamma$. Since $\gamma > 1$, this happens outside the infinite element domain. The pole distance, Δ_p, controls just how far away this singularity is located. If Δ_p is small compared to the scaled width, Δ_s, the coordinate stretching will be very nonlinear, progressing from gentle close to the boundary with the physical domain to abrupt toward to quasi-infinite boundary. Conversely, if the pole distance is large compared to the scaled width, the stretching will be nearly constant across the domain.

The default pole distance is `dGeomChar`, which is small compared to the physical width. Therefore, the coordinate stretching by default exhibits a nearly $1/r$ behavior, which is suitable for making optimal use of mesh resolution when the dependent variable also behaves as $1/r$ for large r, where r is the distance from any sources or inhomogeneities.

> By setting the physical width relatively small and the pole distance large, it is often possible to use the infinite elements also for simulating a large but finite domain.

Infinite Element Domain

An **Infinite Element Domain** node ( ) applies a rational coordinate scaling to a layer of virtual domains surrounding the physical region of interest. When the dependent variables vary slowly with radial distance from the center of the physical domain, the finite elements can be stretched in the radial direction such that boundary conditions on the outside of the infinite element layer are affectively applied at a very large distance from any region of interest.

To add an Infinite Element Domain to any Component, on the **Definitions** toolbar click **Infinite Element Domain**.

COORDINATE SYSTEM IDENTIFIER

The default **Identifier** is `ie1`. The identifier provides a namespace for variables created by the **Infinite Element Domain** node. For example, the scaled x coordinate can typically be accessed in equations and postprocessing as `ie1.x`. See the **Equation View** subnode for a complete list of available variables.

> To display the **Equation View** node under all nodes creating variables, click the **Show** button and select **Equation View**. See also Equation View.

DOMAIN SELECTION

Select a set of domains conforming to the selected geometry type. See Standard Geometry Configurations.

GEOMETRY

Select a **Type**—**Cartesian** (the default), **Spherical**, or **Cylindrical**.

- If **Spherical** is selected, enter the position of the center of the spherical geometry in the **Center coordinate** table. For axisymmetric models, only the z coordinate is required since the geometry must be centered on the axis.
- If **Cylindrical** is selected, enter the position of a point on the cylinder axis in the **Center coordinate** table. For 3D models, also enter a **Center axis direction** vector.

SCALING

Enter expressions for the **Physical width** (SI unit: m) and the **Pole distance** (SI unit: m). The default values, `1e3*dGeomChar` and `dGeomChar`, respectively, leads to an infinite element domain which is very large compared to the geometry dimensions and with a nearly singular $1/r$ stretching.

Known Issues When Modeling Using Infinite Elements

Be aware of the following when modeling with infinite elements:

USE OF ONE SINGLE INFINITE ELEMENTS NODE

Use a separate Infinite Elements node for each isolated infinite element domain. That is, to use one and the same Infinite Elements node, all infinite element domains must be in contact with each other. Otherwise the infinite elements do not work properly.

ELEMENT QUALITY

The coordinate scaling resulting from infinite elements also yields an equivalent stretching or scaling of the mesh that effectively results in a poor element quality. (The element quality displayed by the mesh statistics does not account for this effect.)

The poor element quality causes poor or slow convergence for iterative solvers and makes the problem ill-conditioned in general. The *vector element* formulations (the ones using two or more components of a vector field variable) in the AC/DC Module are particularly sensitive to low element quality.

For this reason, it is strongly recommended to use swept meshing in the infinite element domains. Select the sweep direction to be the same as the direction of scaling. For Cartesian infinite elements in regions with more than one direction of scaling it is recommended to first sweep the mesh in the domains with only one direction of scaling, then sweep the domains with scaling in two directions, and finally sweep the mesh in the domains with infinite element scaling in all three directions.

COMPLICATED EXPRESSIONS

The expressions resulting from the stretching get quite complicated for spherical infinite elements in 3D. This increases the time for the assembly stage in the solution process. After the assembly, the computation time and memory consumption is comparable to a problem without infinite elements. The number of iterations for iterative solvers might increase if the infinite element regions have a coarse mesh.

ERRONEOUS RESULTS

Infinite element regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the infinite element parameter to give erroneous results.

USE THE SAME MATERIAL PARAMETERS OR BOUNDARY CONDITIONS

The infinite element region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an infinite element region.

References for PMLs and Infinite Element Domains

1. O.C. Zienkiewicz, C. Emson, and P. Bettess, “A Novel Boundary Infinite Element,” *Int. J. for Numerical Methods in Engineering*, vol. 19, no. 3, pp. 393–404, 1983.

2. J.P. Bérenger, “A Perfectly Matched Layer for the Absorption of Electromagnetic Waves,” *J. Comput. Phys.*, vol. 114, pp. 185–200, 1994.

3. Jianming Jin, *The Finite Element Method in Electromagnetics*, 2nd ed., Wiley-IEEE Press, 2002.

6

Visualization and Selection Tools

COMSOL Multiphysics® provides a number of tools to visualize and control how you view models and select parts of the model geometry in the Graphics window and the settings windows.

Working with Geometric Entities

The topics in this section provide you with an introduction to the following:

- About Geometric Entities defines the types of geometric entities in COMSOL, including information about adjacent and overlapping objects in the Graphics window.
- The Graphics Window is the environment where the geometry is visualized and selected.
- About Selecting Geometric Entities is an important section that helps you to understand how to highlight, select, or hide any part of the geometry using buttons, mouse clicks, keyboard shortcuts, or combinations of actions. Many physics feature node settings windows have a common selection section, The Geometry Entity Selection Sections, which also has several useful buttons available on the Settings Window Toolbar.
- The Selection List Window is a tool to help you to list all the specific geometric entities in the model and to locate and select, for example, small parts of complex geometries.
- About Highlighted Geometric Entities in the Graphics Window describes the color scheme used to help you visually determine what geometric entities are included or excluded in a model. This makes it easy to add or remove domains, boundaries, edges, or points to the model.
- The section Selecting and Clearing Selection of Geometric Entities has a table with a list of the different ways to select geometry using a variety of windows, mouse buttons, clicks, and keyboard shortcuts.
- The Graphics Window Toolbar Buttons includes a table with the different icons that display in the Model Builder (based on space dimension). The rest of the section describes the tasks related to the toolbar—Zooming In and Out in the Graphics Window, Changing Views in the Graphics Window, Moving Around and Rotating 3D Geometry, Lighting, Transparency, and Wireframe Rendering, and Hiding and Showing Geometric Entities.

About Geometric Entities

Conceptually, a geometry is a collection of bounded *geometric entities*. Those entities are volumes, surfaces, curves, or points. Geometric entities include *domains*, *boundaries*, *edges* (3D only), and *points*. For example, a 3D cube consists of one domain with six boundaries. The six boundaries have 12 edges and the edges connect at eight points (see Figure 6-1). This enables visualization of a cube by displaying one or more of these four types. For instance, you can create a wireframe plot by rendering only the cube's edges.

Geometric entities of the maximum dimension are called *domains*, while those of the next highest dimension are called *boundaries*. The boundaries are sometimes referred to as *faces* in 3D and *edges* in 2D. The *vertices* are also called *points*.

Table 6-1 summarizes the terms used in COMSOL Multiphysics.

TABLE 6-1: NAMES OF GEOMETRIC ENTITIES IN DIFFERENT SPACE DIMENSIONS

ENTITY DIMENSION	NAME IN 3D	NAME IN 2D	NAME IN 1D	NAME IN 0D
3D	domain			
2D	boundary	domain		
1D	edge	boundary	domain	
0D	vertex	vertex	boundary	domain

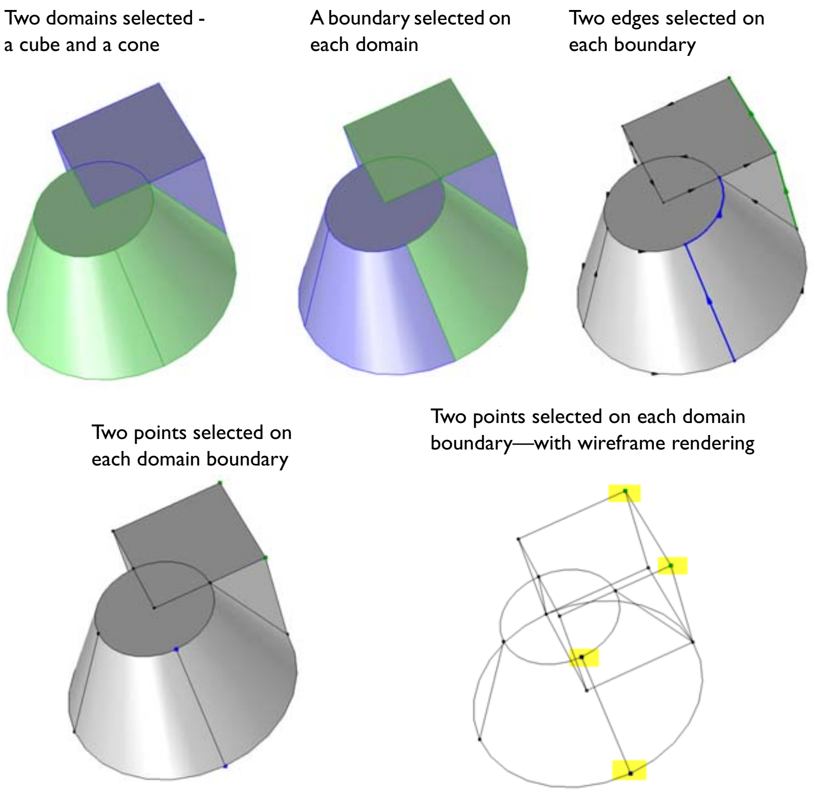
These rules apply to domains:

- The (interiors of the) domains are disjointed. However, this is only strictly true if the finalization method is to form a union. When it is to form an assembly, domains can overlap (though that is normally considered a modeling error).
- Every geometric entity is bounded by entities of smaller dimension. In particular, a domain (in 3D, 2D, or 1D) is bounded by boundaries, edges (in 3D), and vertices (in 3D and 2D). A boundary (in 3D or 2D) is bounded by edges (in 3D) and vertices. An edge is bounded by vertices.

ADJACENT, OVERLAPPING, AND HIDDEN OBJECTS

Geometry objects are *adjacent* if they connect directly to each other. Hence all boundaries, edges, and points on the cube are adjacent to the domain. An edge on the cube is adjacent to two boundaries and two points.

When you click a 3D geometry comprised of several objects, geometric entities of the same type might overlap and hide each other at the point where you click. *Overlapping* objects, such as interior boundaries, are selected starting with the closest geometric entity and ending with the entity the farthest away. Use the scroll wheel (mouse wheel) to move the highlighting from the closest overlapping entity forward and back by rolling the wheel forward and backward (if you use COMSOL on a computer with a touchpad instead of a mouse, use its equivalent to the wheel; for example, moving two adjoining fingers up or down the touchpad). If you can move the mouse wheel in small distinct increments, each such increment moves the selection to the next or previous entity that you can reach. Alternatively, use the up arrow and down arrow keys to select the next or previous entity, respectively.



Two domains selected - a cube and a cone

A boundary selected on each domain

Two edges selected on each boundary

Two points selected on each domain boundary

Two points selected on each domain boundary—with wireframe rendering

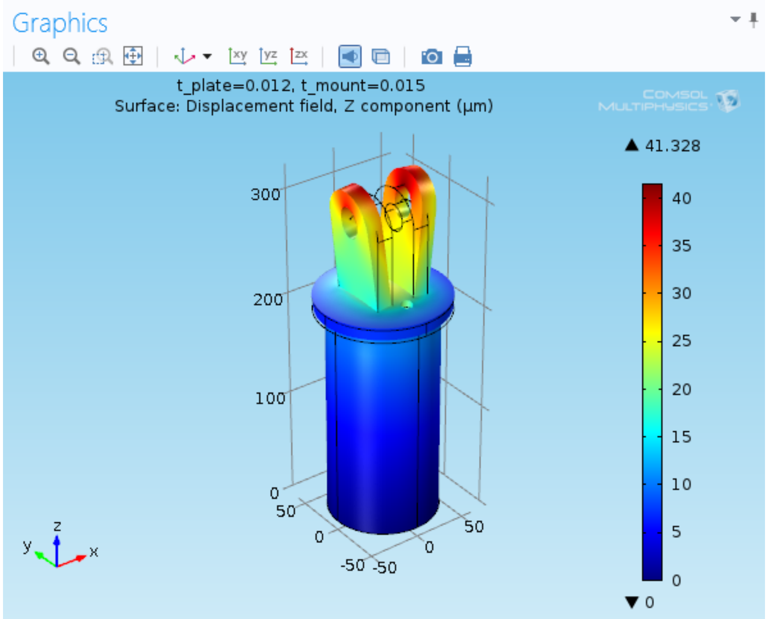
Figure 6-1: A 3D geometry comprises domains, boundaries, edges, and points.

- The Graphics Window
- About Selecting Geometric Entities
- About Highlighted Geometric Entities in the Graphics Window

The Graphics Window

The **Graphics** window (Figure 6-2) is a graphical view of the geometry, mesh, and results of the model. The window has useful tools for changing the view and selecting multiple entities—geometry objects when creating the geometry as well as domains, boundaries, edges, and points to define the physics or to select geometric entities for fine-tuning the mesh or evaluating quantities in a certain part of the model, for example.

The toolbar at the top of the Graphics window has a set of tools for changing the visualization (for example, to zoom in or out or to add transparency) and for making selections. The available tools are dynamic and change based on the space dimension and what you are viewing in the graphics window at the time.



t_plate=0.012, t_mount=0.015
Surface: Displacement field, Z component (µm)

Figure 6-2: The Graphics window displaying a Surface plot for the Diagonal Mounting Detail of a Communication Mast model. This image displays for Windows users. The Mac and Linux Graphics window looks slightly different but functions the same and has the same toolbar.

- Capturing and Copying Screen Shots
- About Geometric Entities
- The Graphics Window Toolbar Buttons
- Named Selections

Basic Selection Concepts

The following basic selection concepts are useful for picking geometric entities where you want to apply physics, boundary conditions, or other parts of the model:

- To pick a geometric entity in the **Graphics** window and add it to the current node's selection list, simply click it. The geometric entity then appears in the selection list. Click again to deselect.
- You can also use the **Selection List** window to pick geometric entities, which you then add to the current selection by right-clicking and selecting **Add to Selection** or clicking the corresponding toolbar button (➕ . You can select multiple entities from the selection lists using Ctrl-click or Shift-click.
- It is good practice to add selection nodes with a selection of geometric entities that you can give a descriptive name and that contains geometric entities that represent a specific part of the geometry (a ground plane or an outlet, for example). You can then choose that selection node as a predefined selection from the **Selection** lists in the physics nodes, for example.
- For adding physics nodes from the toolbars, it can be useful to use a *preselection* by setting the Active button to off in the current settings window. You can then select geometric entities and click the toolbar button for the physics that you want to add. That physics node, when added, then gets the selection that you have preselected.

> You can choose to use the selection mechanisms and highlighting that was available in earlier versions of COMSOL by selecting the **Use classic selections** check box on the **User interface** page in the **Preferences** dialog box (and restarting COMSOL).

About Highlighted Geometric Entities in the Graphics Window

COMSOL highlights geometric entities at different stages of selection. A geometric entity is highlighted in red, blue, green, yellow, or with no highlight (gray) to indicate its status.

> Another visual cue to help you work in the Graphics window is that the geometry can have thicker edges (with OpenGL and Software rendering only, not DirectX), or larger points to highlight the different geometric entities selected. With the default graphics preference, to optimize for quality (for graphics cards that support it), highlighted geometric entities appear with a "glowing edge." If required, change the default from The Preferences Dialog Box under **Graphics and Plot Windows>Visualization**.

BLUE

A geometric entity highlighted in blue is included in the selection list for the specific node. See Figure 6-4 for an example.

When assigning geometric entities to a node and you hover over that entity in the Graphics window, click once to add it to a selection list. See Figure 6-5 for an example.

RED

When a geometric entity is selected in a selection list or on The Selection List Window, it is highlighted in red to help you locate it on the Graphics window. See Figure 6-4 for an example.

When assigning geometric entities to a node, and you hover over that entity in the Graphics window, it displays in red to indicate it has not been added to the selection. See Figure 6-5 for an example.

GREEN

When you hover over a geometric entity in the Graphics window, it displays in green to indicate that it is included in the selection list. See Figure 6-4 for an example.

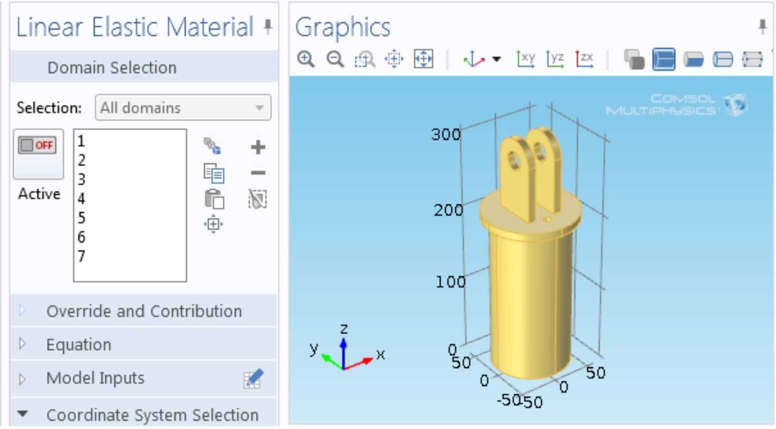
GRAY (NO HIGHLIGHT)

If the geometric entity is gray (that is, not highlighted), it means it is not selected or included for that node and geometric entity level. See Figure 6-5 for an example.

YELLOW

Next to the settings window's selection list there is an **Active** button to toggle between turning ON and OFF selections; that is, making the selections active for that selection list. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

When the button is toggled to OFF, the selection mode is a *preselection* that will be used for a selection that you add to the model from the toolbar, and the selection for the current node in the model tree is highlighted in yellow in the **Graphics** window. See Figure 6-3.

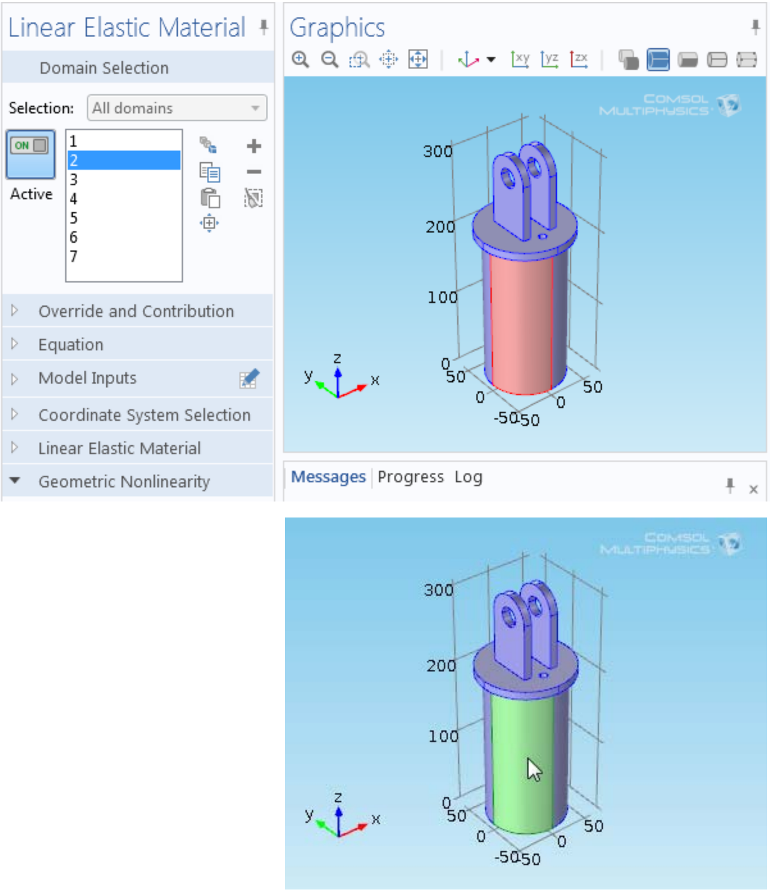


Selection: All domains

Figure 6-3: When the Active button is OFF, the selection for the current node is highlighted in yellow.

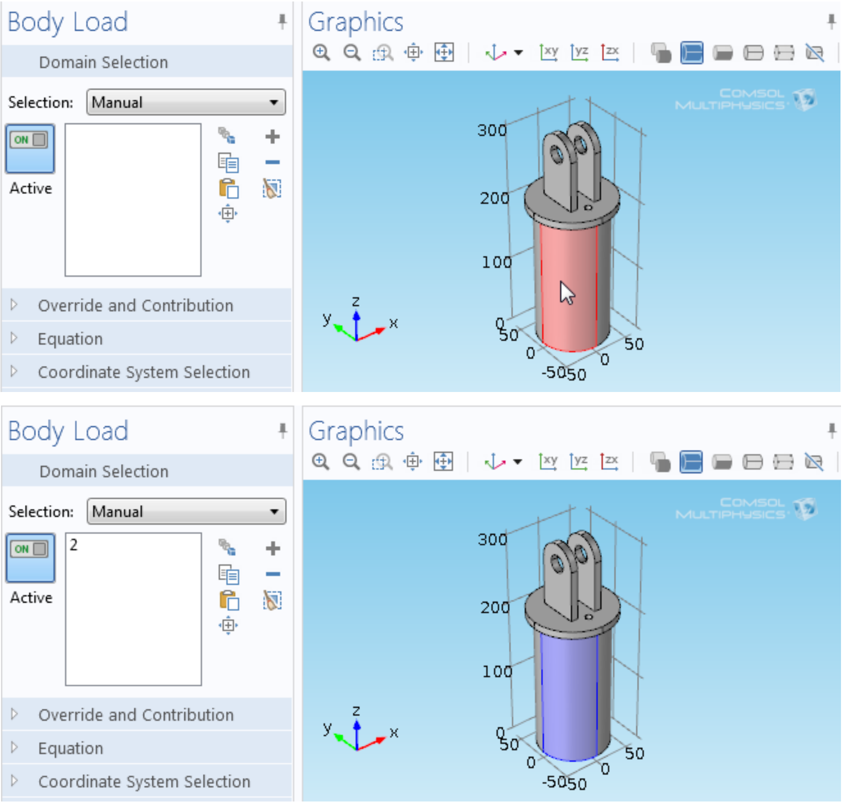
Hover over an entity in the Graphics window. When the selection is active, add or subtract it from the selection list (see Figure 6-6) by observing how the colors cycle on and off:

COLOR WHEN HOVERED OVER	CURRENT STATUS	NEW COLOR WHEN CLICKED	NEW STATUS WHEN CLICKED
Green	Included as a selection for the node.	Gray	Not included as a selection for the node.
Red	Not included as a selection for the node.	Blue	Included as a selection for the node.



Selection: All domains

Figure 6-4: Domain 2 is selected on the Linear Elastic Material settings window. Domain 2 is highlighted in red in the Graphics window to indicate where it is on the geometry (top image). Hover over a geometric entity in the Graphics window and it displays in green (bottom image). The blue highlight indicates that these domains are included as the selection, in this case because it is a default node and the selection cannot be changed. This example uses the Diagonal Mounting Detail of a Communication Mast model.


Body Load

Domain Selection

Selection: Manual

Active

Override and Contribution

Equation

Coordinate System Selection

Body Load

Domain Selection

Selection: Manual

2

Active

Override and Contribution

Equation

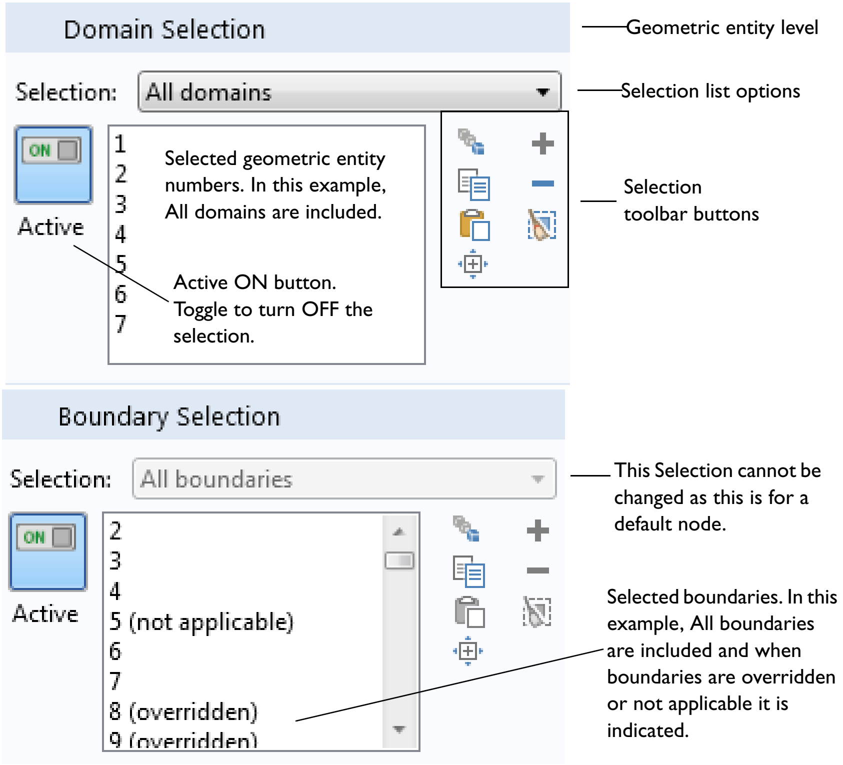
Coordinate System Selection

Figure 6-5: You want to add a domain to the Body Load node to make the load act on that domain. When you hover over a geometric entity in the Graphics window it displays in red to indicate it has not been added to the selection list (top). Click the domain to add it to the selection list on the settings window and it displays in blue to indicate it has successfully been added (bottom). In both images, gray-colored domains are not selected or included.

About Selecting Geometric Entities

Throughout COMSOL there are many lists of selected geometric entities, all based on the same principle—pick a domain, boundary, edge, or point and use methods to add or remove these geometric entities to create *selections* that define, for example, the parts of the geometry where a material or boundary condition is active. Such lists appear in settings windows for defining equations and material properties, boundary conditions, sources, and other parts of the model's physics, or the Variables node (a=) definitions for variables that are not defined in the entire model.

All levels of geometry can be treated individually. You can add and remove 3D geometric entities (domains, boundaries, edges, or points) to selection lists in different ways, including buttons on the Graphics toolbar (The Graphics Window Toolbar Buttons), using The Selection List Window, clicking directly on the geometry, or clicking buttons in the settings window. Table 6-2 lists the buttons that display on every settings window with a geometric entity selection list as displayed in Figure 6-6.



Selection: All domains

1
2
3
4
5
6
7

Selection: All boundaries

2
3
4
5 (not applicable)
6
7
8 (overridden)
9 (overridden)

Geometric entity level

Selection list options

Selection toolbar buttons

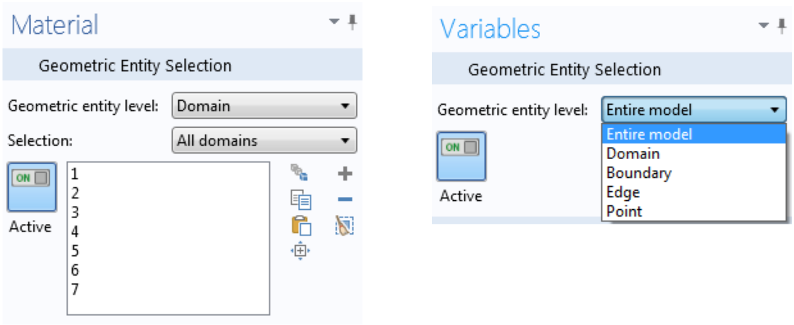
Selected geometric entity numbers. In this example, All domains are included.

Active ON button. Toggle to turn OFF the selection.

This Selection cannot be changed as this is for a default node.

Selected boundaries. In this example, All boundaries are included and when boundaries are overridden or not applicable it is indicated.

Figure 6-6: The selection list and toolbar on a physics node settings window. The geometric entity level for the top settings window is domains. The bottom settings window is for a default boundary condition node where the selection cannot be changed, although a boundary can be overridden.



Geometric entity level: Domain

Selection: All domains

1
2
3
4
5
6
7

Geometric entity level: Entire model

Figure 6-7: The Geometric entity level selection list and toolbar on a Material node settings window (left) and the Variables settings window (right).

In The Graphics Window the geometric entities are color highlighted as you make the selections, and you can lock the selections by turning off the active selections or click the **Select Node** button (in 3D) in the **Graphics** window toolbar.

THE GEOMETRY ENTITY SELECTION SECTIONS

The name of the section where the list of selected geometric entities is managed depends on the geometric entity level. For example, Figure 6-6 displays a **Domain Selection** section.

Every geometric entity selection section also has an **Active** button to toggle between turning ON and OFF active selections for that node. For Windows users, the buttons are ON and OFF. For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

For physics nodes that are default nodes (see Physics Default Nodes) the selection defaults to all geometric entities on the applicable level (all domains or all boundaries, for example, in Figure 6-6), and the **Selection** list is not active. You can add other nodes that override the default nodes for some or all entities. Those entities are then marked **(overridden)** in the selection list for the default node.

- **Geometric Entity Selection**: For Materials and Variables nodes, where you first select the level (domain, boundary, and so on), from a **Geometric entity level** list. See Figure 6-7 for an example using the “Diagonal Mounting Detail of a Communication Mast” model.
- **Domain Selection**: For nodes that define, for example, material models, sources, and body loads in domains. See Figure 6-6 for an example.
- **Boundary Selection**: For nodes that define, for example, boundary conditions.
- **Edge Selection**: For nodes that define, for example, conditions and forces on edges. This is applicable to 3D models only.
- **Point Selection**: For nodes that define, for example, point sources and point loads.

From the **Selection** list you can choose one of the following options:

- **Manual** (the default): Select the geometric entities directly in the **Graphics** window, using The Selection List Window, or using the **Paste Selection** button. See below for more information about those selection methods.

If you start by setting the **Geometric entity level** to **Domain**, and then select **All domains**, the **Selection** list displays all domains. If you make any changes to this list (for example, remove a domain) the **Selection** list reverts to **Manual**.

- **All domains**, **All boundaries**, **All edges**, **All points**: Depending on the geometric entity level, you can choose one of these options to select all entities. See Figure 6-6 for example.
- Defined named selections: Selection nodes added in the geometry sequence or under **Definitions** (as well as selections created from Boolean operations, for example) are available in the **Selection** lists for nodes that define model properties for the same geometric entity level. You can rename such selection nodes to better reflect what the selected entities represent. A named selection can consist of, for example, the domains where a volume force acts, the boundaries where an inflow occurs, or points that are grounded. Named selections are useful for reusing selections in a model and to clearly indicate what parts of geometry that the selected entities include or represent. See Creating Named Selections.

In the lists of selected entities, **(overridden)** and **(not applicable)** can display next to the label (the number) of a selected entity. See Figure 6-6 and Physics Node Status for more information about these status indicators. There is also an **Override and Contribution** section in all physics nodes. It provides an overview of how the physics nodes and their selections interact. See Physics Exclusive and Contributing Node Types.

Pair Selection

If the geometry is an assembly you have access to specific pair conditions (typically on boundaries) that you choose from a **Pairs** submenu on the main physics nodes’ context menus. In the settings window for such pair nodes, a **Pair Selection** section contains a list of the applicable pairs (typically identity pairs). The **Boundary Selection** section (or another standard selection section) is then unavailable and shows the entity numbers for the boundaries, for example, that the selected pairs include. You specify for which pairs the pair condition is applied by selecting on or more pairs from the **Pairs** list. See Identity and Contact Pairs.

📄	See Table 6-3 for the many different ways to select geometric entities using toolbar buttons, mouse click options, page settings, and keyboard shortcuts.

SETTINGS WINDOW TOOLBAR

TABLE 6-2: GEOMETRIC ENTITY SELECTION BUTTONS ON VARIOUS SETTINGS WINDOWS

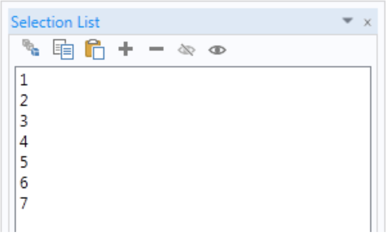
BUTTON	NAME	DESCRIPTION
ON □ (Windows) ⏻ (Mac and Linux)	Active ON	Click the Active button ON to make the selection of geometric entities to the Selection list active for that particular node.
□ OFF (Windows) ⏻ (Mac and Linux)	Active OFF	Click the Active button OFF to lock/deactivate the geometric entities in the Selection list for that particular node. The selected entities in the geometry are highlighted in yellow in the Graphics window. See Figure 6-3 for an example. The Graphics window is then available for preselection of entities for a new node.
	Create Selection	Use this button to create selection nodes under Definitions to represent various parts of the geometry and simplify the process of assigning materials, model equations, boundary conditions, and model properties. See Creating Named Selections. This button is also on The Selection List Window.
	Copy Selection	Use this button to copy the selection from the list in the settings window to the clipboard. See Copying and Pasting Selection Lists. This button is also on The Selection List Window.
	Paste Selection	Use this button if you have a list of geometric entities in a file or document that you want to type in and then paste into a selection list. If you copy a selection from a document to the clipboard, you can paste the selection directly using Ctrl+V. An example is a list of geometric entity numbers described as a step in a modeling instruction. See Copying and Pasting Selection Lists. This button is also on The Selection List Window.
+	Add to Selection	Use this button to add a geometric entity to a selection list. See Table 6-3. This button is also on The Selection List Window.
−	Remove from Selection	Use this button to remove a geometric entity from the Selection list. See Table 6-3. This button is also on The Selection List Window.
	Clear Selection	Use this button to clear a selection. Clearing the selection also activates the selection for that node.
⊞	Zoom Selected	Use this button to zoom into the selected geometric entities.

The Selection List Window

Use the **Selection List** window (see Figure 6-8) to make it easier to choose objects, for example, while working with complex geometries and when you need to easily locate a geometric entity that is not easily viewed. The Selection List is particularly useful when you know the geometric entity number to select; for example, when you are following step-by-step model instructions to build a model (in that case you can also copy and paste the selections directly from the instructions).

Win	To open the window, from the **Home** ribbon select **More Windows>Selection List** ( ).

Mac Linux	To open the window, select **Windows>Selection List** ( ).


Selection List

1
2
3
4
5
6
7

Figure 6-8: The Selection List window and toolbar. Most of the toolbar buttons are also on a node settings windows. In this example, the numerical representation for the domains is listed. See Table 6-2 *for button information.*

WORKING WITH THE SELECTION LIST WINDOW

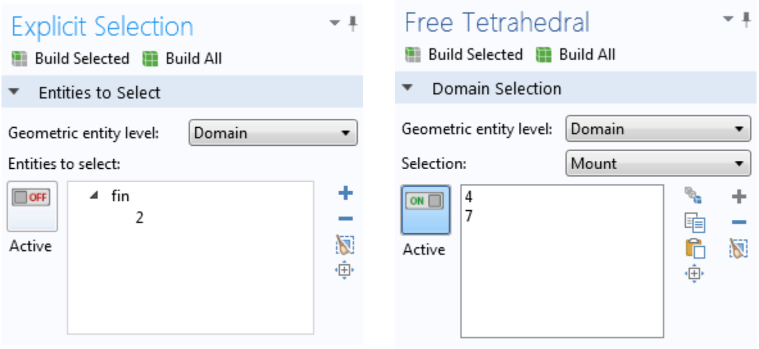
The Selection List window displays all geometric entities of a certain type (boundaries, for example). COMSOL Multiphysics determines the geometric entities listed based on where in the model you are working.This is different from selection lists in settings windows, which contain lists of the selected entities only (see About Selecting Geometric Entities, Figure 6-6).

Click any item to see it highlighted in The Graphics Window—except if the item is hidden, which is indicated in the Selection List by **(hidden)**—and select items as described in Selecting and Clearing Selection of Geometric Entities. For example, use the Selection List in these situations:

Materials, Physics, and Boundary Conditions: When working in windows with Selection or Geometric scope sections (a **Selection** window under a **Definitions** node for example), or anywhere you assign materials, physics, boundary conditions, and other Component settings. The Selection List displays the specific *geometric entity level* selected (domain, boundary, edge, or point). See Figure 6-13 for an example.

Geometry: When in the Model Builder under the **Geometry** node—the geometry objects are displayed in the **Entities to select**, for example, `ext1` (extrusion), `blk1 (solid)`, or `cone1 (solid)` (Figure 6-9). You might also use it with a Chamfer or Fillet geometry feature when you want to locate specific points. To specify the selection level, click the **Select Points** button in the Graphics toolbar and add the points to the **Vertices to fillet** or **Vertices to chamfer** lists. See Creating Named Selections in the Geometry Sequence for details about creating selections based on geometry sequences.

Meshing: When in the Model Builder under the **Mesh** node, the list also includes information on which entities are meshed by adding **(meshed)** to the right of the meshed entities. If the Geometry has Mesh Construction entities, the list also specifies if a construction entity has been removed; see Mesh Control Entities. This is indicated next to the entity in the list by **(meshed and removed)**.



Explicit Selection

Build Selected | Build All

Entities to Select

Geometric entity level: Domain

Entities to select:

OFF Active

fin
2

Free Tetrahedral

Build Selected | Build All

Domain Selection

Geometric entity level: Domain

Selection: Mount

ON Active

4
7

Figure 6-9: An example of an Explicit Selection settings window for a Geometry sequence (left) and a Free Tetrahedral settings window for a Mesh node (right).

- Creating Named Selections
- Named Selections
- About Selecting Geometric Entities

Selecting and Clearing Selection of Geometric Entities

The sections About Selecting Geometric Entities and The Selection List Window give an overview of some of the tools and windows available to highlight and select geometric entities. About Highlighted Geometric Entities in the Graphics Window describes the different colors that display in the Graphics window to help you select geometric entities to include in your model.

Table 6-3 contains the description of the different ways to complete the same task of selecting (or deselecting) geometric entities.

TABLE 6-3: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Select any level of geometry:	In The Graphics Window, click the geometric entity to add it to an active selection list in the current node's Settings window. Open The Selection List Window and click entity numbers (or names); then click the **Add to Selection** button (+) to add it to an active selection list in the current node's Settings window.
Select all parts of the geometry:	In most settings windows' selection sections, the option to select **All domains**, **All boundaries**, **All edges**, or **All points** adds all geometric entities of that type to the list of selected entities. The selected items are highlighted in the **Graphics** window and all entities are kept selected even if the geometry changes. Open The Selection List Window, use the Shift or Ctrl keys to select all the entity names (or numbers). You can also click the main geometry node to select all entities that the node includes. Then click the **Add to Selection** button (+) to add it to an active selection list in the current node's Settings window. In the **Graphics** window, click the **Select All** button (⬚ in 3D), or click outside of the geometry or press Ctrl+A to select all entities. For Windows users, a **Select All** button is available on a customized Quick Access Toolbar. For Mac and Linux users, from the main menu select **Edit>Select All**. This highlights and selects all entities but does not confirm the selection or lock it if the model changes.
Clear the selection of all parts of the geometry not added to a selection list:	In the **Graphics** window, click the **Clear Selection** button (⬚), or click outside of the geometry or press Ctrl+D to clear all selected entities. For Windows users, a **Clear Selection** button is available on a customized Quick Access Toolbar. For Mac and Linux users, from the main menu select **Edit>Clear Selection**.
Move, rotate, and then select (3D only):	Multiple mouse actions can be done together. For example, use the mouse to rotate or move the object left and right to locate the geometric entity to add to a selection list; then click to add to the selection.
Lock the geometry during selection:	In the **Graphics** window, click the **Select None** button (⊘). Then no clicks in the graphics highlights or picks any geometric entities, so you can move and rotate the geometry freely.
Add to Selection + :	In the **Graphics** window, click a red highlighted entity to add it and make it blue. Or select one or more geometric entities and click the **Add to Selection** button. In the **Selection List** window, select the entity names to add, and click the **Add to Selection** button. You can also paste selections from a file. See Copying and Pasting Selection Lists for information. For user-defined selections this action must be completed on the selection page. See Create an Explicit Selection from the Selection List Window for information.

TABLE 6-3: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Select Box: 3D 2D 1D	To select multiple parts of the geometry, in the **Graphics** window, click the **Select Box** button then click and hold the left mouse button to draw a square over the geometry. It is like a “rubberband” selecting all entities enclosed by this operation. The selected geometric entities are added to the selection list. In the **Selection List** window, use the Shift or Ctrl keys to select the entity names (or numbers). Click the **Add to Selection** button.
Select Objects: 3D 2D 1D	In the **Graphics** window, click the **Select Objects** button. Click to select the geometry object and add it to the selection list.
Select Domains: 3D 2D 1D	In the **Graphics** window, click the **Select Domains** button. Click to select the domains and add it to the selection list. In the settings window, select **Domain** from a **Geometry entity level** list and then click in the **Graphics** window. Only domains are highlighted.
Select Boundaries: 3D 2D 1D	In the **Graphics** window, click the **Select Boundaries** button. Click to select the boundary and add it to the selection list. In the settings window, select **Boundary** from a geometry entity level selection list and then click in the **Graphics** window. Only boundaries are highlighted.
Select Edges :	In the **Graphics** window, click the **Select Edges** button. Click to select the edge and add it to the selection list. In the settings window, select **Edge** from a geometry entity level selection list and then click in the **Graphics** window. Only edges are highlighted.
Select Points: 3D 2D	In the **Graphics** window, click the **Select Points** button. Click to select the point and add it to the selection list. In the settings window, select **Point** from a geometry entity level selection list and then click in the **Graphics** window. Only points are highlighted.
Select None: 3D 2D 1D	In the **Graphics** window, click the **Select None** button to turn off all selections so that you can click in the Graphics window to move or rotate the geometry, for example, without adding any object or entity to a selection.
Select All :	In the **Graphics** window, click the **Select All** button to select all objects or entities.
Edit :	Interactive editing of 2D geometry objects so that you can click-and-drag to resize and move geometry objects. Alt-click a 2D geometry object to edit its underlying properties and drag its control points, for example. Right-click to edit that geometry editing mode. See Moving and Scaling 2D Objects in the Graphics Window.

TABLE 6-3: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Remove from Selection ▬:	In the **Graphics** window, highlight or select the geometric entity and click the **Remove from Selection** button. In the **Graphics** window, click to highlight and select the geometric entity to remove. Any blue geometric entity turns gray to indicate it is removed from the selection. In the **Selection List** window, use the Shift or Ctrl keys to select the entity names to remove. Click the **Remove from Selection** button. For user-defined selections this action must be completed on the selection page. See Create an Explicit Selection from the Selection List Window for information.
Clear Selection ⊘:	Click the **Clear Selection** button to clear all selections from the selection list. For user-defined selections this action must be completed on the selection page. See Create an Explicit Selection from the Selection List Window for information.

The Graphics Window Toolbar Buttons

Some of the toolbar buttons available on the **Graphics** window are different based on the space dimension of the Component. The buttons also correspond to domain, boundary, edge, and point level nodes that display under the physics, which are also based on the Component space dimension as shown in Table 6-4. The buttons in Table 6-5 are available in any space dimension.

In the tables there are links to the corresponding sections that contain instructions about how to do the listed tasks.

TABLE 6-4: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

NAME	3D	2D AND 2D AXISYMMETRIC	1D AND 1D AXISYMMETRIC
Selecting and Clearing Selection of Geometric Entities			
Select Boundaries	▬	▭	─▪
Select Box	⬚	⬚	⬚
Select Domains	■	■	─
Select Objects	■	■	═
Select Points	⬚	⬚	—
Select Edges	⊟	—	—
Select None	⊘	⊘	⊘
Select Edit	—	✎	—
Changing Views in the Graphics Window			
Go to XY View	xy	—	—
Go to YZ View	yz	—	—
Go to ZX View	zx	—	—
Lighting, Transparency, and Wireframe Rendering			

TABLE 6-4: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

NAME	3D	2D AND 2D AXISYMMETRIC	1D AND 1D AXISYMMETRIC
Scene Light		—	—
Transparency		—	—
Wireframe Rendering		—	—
Drawing on a 2D Work Plane in 3D			
Align with Work Plane*	—		—
Work Plane Clipping*	—		—

*These buttons are available when using a Work Plane to define a 2D object in 3D.

TABLE 6-5: GRAPHICS TOOLBAR BUTTONS AVAILABLE FOR ALL SPACE DIMENSIONS

BUTTON	NAME	SEE ALSO
	Select All (Ctrl+A)	
	Clear Selection (Ctrl+D)	
	Select and Hide	Hiding and Showing Geometric Entities
	View Unhidden	
	View Hidden Only	
	View All	
↻	Reset Hiding	
🔍	Zoom In	Zooming In and Out in the Graphics Window
🔍	Zoom Out	
	Zoom Selected	
	Zoom Extents	
	Go to Default View	Changing Views in the Graphics Window
📷	Image Snapshot	Capturing and Copying Screen Shots
🖨	Print	Printing from the COMSOL Desktop

ZOOMING IN AND OUT IN THE GRAPHICS WINDOW

BUTTON	NAME	ACTION
⊕ and ⊖	Zoom In and Zoom Out	Click the **Zoom In** button to zoom in. Click the **Zoom Out** button to zoom out. *3D only*: Click the middle mouse button and drag it forward and backward to zoom in and out of the object. The zoom is centered where the first click is made in the **Graphics** window.
⬚	Zoom Box	To zoom into a general area of the geometry, click the **Zoom Box** button then click and drag to highlight a section of the geometry to zoom into.
⊞	Zoom Selected	Click the **Zoom Selected** button to zoom into the selected geometric entities. This button is also available in connection with the selection lists for domains, boundaries, edges, and points.
⊞	Zoom Extents	Click the **Zoom Extents** button to zoom out and fit the complete geometry into the window.

CHANGING VIEWS IN THE GRAPHICS WINDOW

BUTTON	NAME	ACTION
xy yz zx	Go to XY View, Go to YZ View, and Go to ZX View (3D only)	Click the **Go to XY View**, **Go to YZ View**, and **Go to ZX View** buttons to change the view to the xy-, yz-, or zx-plane. The first click selects the plane view with a positive normal direction. A second click on the same button switches to a negative normal direction.
⊥	Go to Default View	Click the **Go to Default View** button to change the view to the default.
	Display a user-defined view:	After creating a **View** under the **Definitions** node, click the down arrow next to the **Go to View** button (⊥) and select a user-defined view from the list.

MOVING AROUND AND ROTATING 3D GEOMETRY

TASK	ACTION AND RESULT	OPERATION ORDER
Rotate the geometry about the axes	In the **Graphics** window, left-click and hold down the mouse button while dragging it in any direction. This rotates the scene around the axes parallel to the screen X- and Y-axes with origin in the scene rotation point.	left-click
Move the visible frame on the image plane in any direction	In the **Graphics** window, right-click and hold down the mouse button while dragging it in any direction.	right-click
Zoom in and out around the mouse position where the action started	In the **Graphics** window, click and hold down the middle mouse button and drag the mouse forward or back to zoom in and out.	middle-click

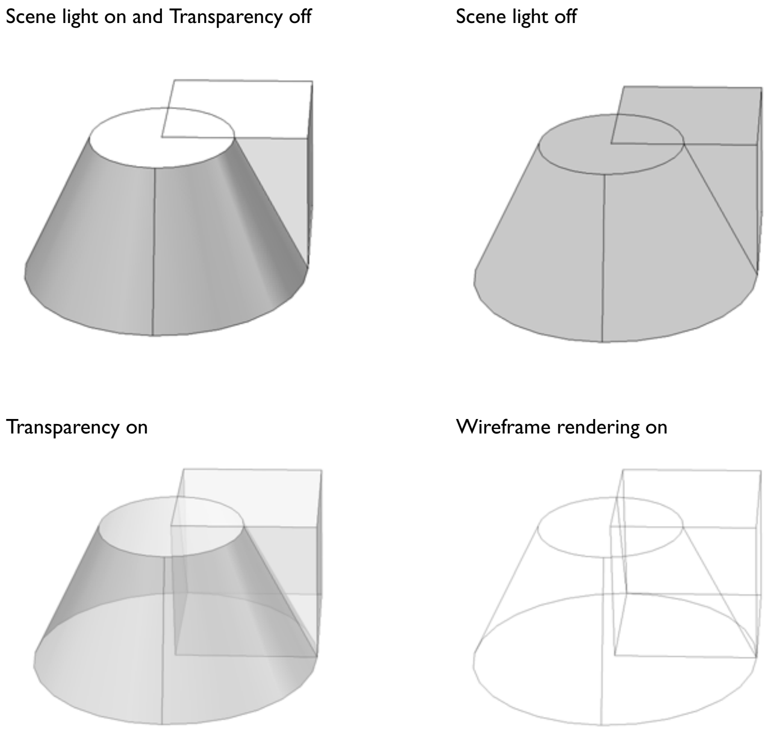
TASK	ACTION AND RESULT	OPERATION ORDER
Rotate about the X- and Y-axes in the image plane (tilt and pan the camera)	Press Ctrl and left-click in the **Graphics** window. While holding down the key and button, drag the mouse in any direction. This places the rotation coordinate system in the camera and rotates around the axes parallel to the screen X- and Y-axes.	Ctrl+left-click
Move the camera in the plane parallel to the image plane	Press Ctrl and right-click in the **Graphics** window. While holding down the key and button, drag the mouse in any direction.	Ctrl+right-click
Rotate the camera about the axis	Press Ctrl+Alt, then left-click in the **Graphics** window. While holding down the keys and button, drag the mouse in any direction. If you have not rotated the camera (using Ctrl+left-click), the effect is the same as when using Alt+left-click.	Ctrl+Alt+left-click
Move the scene in the plane orthogonal to the axis between the camera and the scene rotation point	Press Alt, then right-click the mouse in the **Graphics** window. While holding down the key and button, drag the mouse in any direction.	Alt+right-click
Move the camera into and away from the object (dolly in/out)	Press Ctrl and then click the middle mouse button. While holding down both the key and button, in the **Graphics** window, drag the mouse in any direction.	Ctrl+middle-click
Rotate the camera about its axis between the camera and the scene rotation point (roll direction)	Press Alt, then left-click in the **Graphics** window. While holding down the key and button, drag the mouse in any direction.	Alt+left-click
Move the camera along the axis between the camera and the scene's rotation point	Press Alt, then middle-click in the **Graphics** window. While holding down the key and button, drag the mouse in any direction.	Alt+middle-click

LIGHTING, TRANSPARENCY, AND WIREFRAME RENDERING

The following are available for 3D models.

BUTTON	NAME	ACTION
💡	Scene Light	Click any plot under **Results**. In the **Graphics** window or any other plot window, click the **Scene Light** button to turn it on. Click again to turn scene light off. See Figure 6-10. When creating a **View**, this action toggles the **Scene light** check box on the **View** page.

BUTTON	NAME	ACTION
⧉	Transparency	Click any plot under **Results**. In the **Graphics** window or any other plot window, click the **Transparency** button to turn it on. Click again to turn transparency off. See Figure 6-10. When creating a **View**, this action toggles the **Transparency** check box on the **View** page. See User-Defined Views.
▣	Wireframe Rendering	Click any plot under **Results**. In the **Graphics** window, click the **Wireframe Rendering** button to turn it on. Click again to turn the wireframe off. See Figure 6-10. When creating a **View**, this action toggles the **Wireframe rendering** check box on the **View** page. See User-Defined Views. See also Preferences Settings to set the level of graphic detail to Wireframe and speed up the rendering of complex models or to improve visual appearance.



Scene light on and Transparency off

Scene light off

Transparency on

Wireframe rendering on

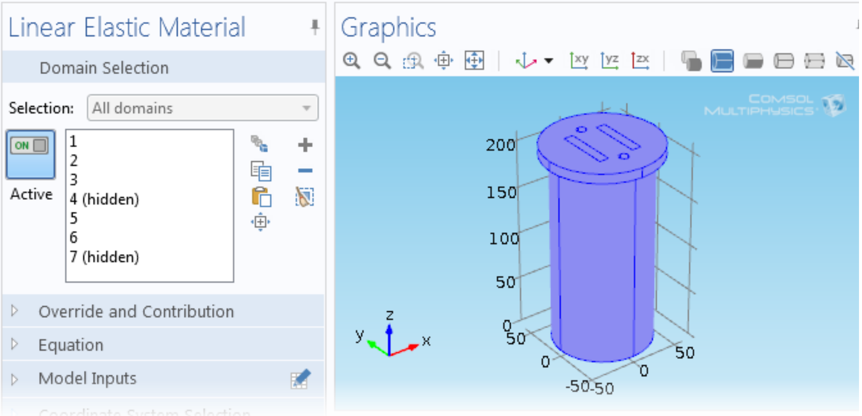
Figure 6-10: Scene light, transparency, and wireframe rendering examples.

HIDING AND SHOWING GEOMETRIC ENTITIES

Selecting an item in any **Selection list** highlights the corresponding geometric entities or objects in the **Graphics** window for selection or deselection. Only the geometric entities you can see in the **Graphics** window are available for selection; that is, hidden objects cannot be selected and selection methods vary based on the Component's space dimension.

BUTTON	NAME	ACTION
	Select and Hide	In the **Graphics** window, toggle the **Select and Hide** button (click to highlight and turn on and click again to turn off). When turned on, click a geometric entity and it is added to the selection list as being a hidden entity. See Figure 6-11. When creating a **View**, right-click the **View** node and select **Hide Geometry Objects**. Select a **Geometric entity level** from the list to hide. Also see Hide Geometry Objects and Hide Geometric Entities when creating a **View**.
	View Unhidden	In the **Graphics** window, click the **View Unhidden** button to display any domains, boundaries, edges, or points not hidden.
	View Hidden Only	In the **Graphics** window, click the **View Hidden Only** button to display only hidden domains, boundaries, edges, or points.
	View All	In the **Graphics** window, click the **View All** button to display all hidden and unhidden domains, boundaries, edges, or points.
	Reset All	In the **Graphics** window, click the **Reset Hiding** button to reset all hidden domains, boundaries, edges, or points to the default. This removes any **Hide Geometry Objects** or **Hide Geometric Entities** subnode added to a **View** node. See Hide Geometry Objects and Hide Geometric Entities.

When the **View Hidden Only** button ( ), **View Unhidden** button ( ), or **View All** button ( ) is clicked on the **Graphics** window toolbar it changes the view accordingly. The selection list on the settings window details what is hidden or shown based on the button clicked. See Figure 6-11 and Figure 6-12 for examples based on the original geometry shown in About Highlighted Geometric Entities in the Graphics Window.



Linear Elastic Material

Domain Selection

Selection: All domains

Active

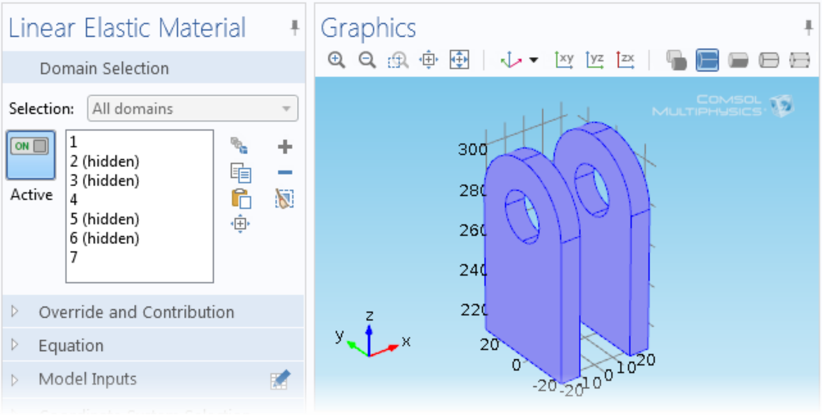
1
2
3
4 (hidden)
5
6
7 (hidden)

Override and Contribution

Equation

Model Inputs

Figure 6-11: An example of the domains that display in the Graphics window when the View unhidden button is clicked. The selection list displays the detail that domains 4 and 7 are hidden in the Graphics window. Compare to Figure 6-12.



Linear Elastic Material

Domain Selection

Selection: All domains

1
2 (hidden)
3 (hidden)
4
5 (hidden)
6 (hidden)
7

Figure 6-12: An example of the domains that display in the Graphics window when the View unhidden only button is clicked. The selection list displays the detail that domains 2, 3, 5, and 6 are hidden in the Graphics window. Compare to Figure 6-11.

Named Selections

This section details how to create named selections to reuse throughout the model when assigning material properties, boundary conditions, and other model settings.

You can create selection nodes under the Component node's Definitions node to represent various parts of the geometry and simplify the process of assigning materials, model equations, boundary conditions, and other model properties. These user-defined selections can be reused during modeling and named using descriptive titles—for example, `Tube`, `Wall`, or `Fluid`. Changes to the selection (for example, by adding or removing a boundary) updates all nodes in the Component that use that particular selection.

Use the buttons listed in Table 6-2 to create, copy, and paste selections. When there is the possibility of overlapping geometric entities, it is recommended that you use The Selection List Window to ensure the correct part of the geometry is selected.

There are different types of selections—Explicit selections, selections by enclosing part of the geometry by a bounding Ball, Box, or Cylinder, Boolean selections (Union, Intersection, Difference, and Complement), and selections of Adjacent geometric entities. To add selection nodes, right-click a **Definitions** node and choose from the **Selections** options as listed in Table 6-6.

You can also right-click the **Geometry** node and choose from **Selections** options similar to those in Table 6-6 for defining selections based on the geometry objects in the geometry sequence. See Creating Named Selections in the Geometry Sequence.

OPEN AN EXAMPLE MODEL WITH DEFINED EXPLICIT SELECTIONS

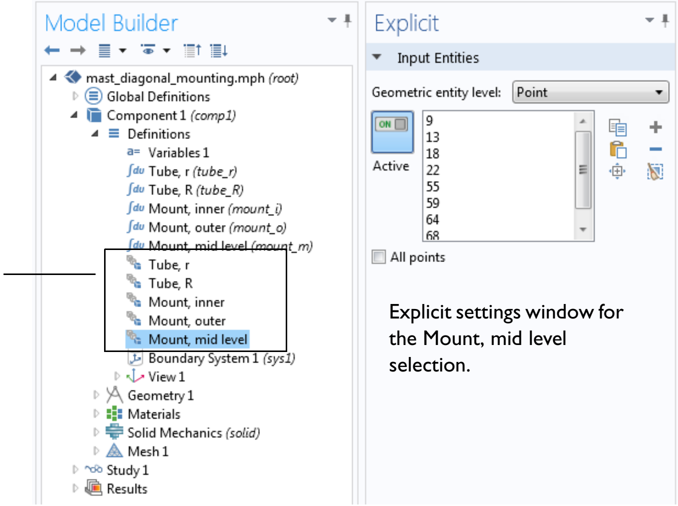
Figure 6-13 uses a COMSOL Multiphysics model library example, which includes several user-defined selections.

1 Open The Model Libraries Window.

2 Navigate to the **COMSOL Multiphysics>Structural Mechanics>mast_diagonal_mounting** model file. Double-click to open it.

3 Expand the **Definitions** node under **Component 1**. Several nodes display in the **Model Builder**. Click the nodes shown in Figure 6-13 to examine the list of geometric entities displayed in the **Explicit** settings window.

In the selection settings windows, also click the **Zoom Selected** (⊞) button to zoom in on the selected geometric entities.



Geometric entity level: Point

9
13
18
22
55
59
64
68

□ All points

Named Explicit selections

Explicit settings window for the Mount, mid level selection.

Figure 6-13: An example of an Explicit selection window for the Mount, mid level node. The Selection nodes under Definitions are renamed by the user.

THE TYPES OF NAMED SELECTIONS

TABLE 6-6: NAMED SELECTIONS BY TYPE

ICON	TYPE	DESCRIPTION
	Adjacent	Use the Adjacent node to create the selection as the adjacent geometric entities (boundaries, for example) to one of more selections.
	Ball	Use the Ball node to create the selection by enclosing part of the geometry by a bounding ball (sphere) to select geometric entities that are partially or completely inside the ball.
	Box	Use the Box node to create the selection by enclosing part of the geometry by a bounding box to select geometric entities that are partially or completely inside the box.
	Cylinder	Use the Cylinder node to create the selection by enclosing part of the geometry by a bounding cylinder to select geometric entities that are partially or completely inside the cylinder.
	Explicit	Use an Explicit node to create the selection using the normal selection tools for individual geometric entities (boundaries, for example) on the geometric entity level chosen.
Union, Intersection, Difference, and Complement		
	Union	Use the Union node to create the selection as the union (addition) of two or more selections.
	Intersection	Use the Intersection node to create the selection as the intersection of two or more selections.
	Difference	Use the Difference node to create the selection as the difference between a set of one or more selections and another set of one or more selections.
	Complement	Use the Complement node to create the selection as the complement (inverse) of one or more selections.

- Grouping Nodes by Space Dimension and Type
- Creating Named Selections in the Geometry Sequence
- Working with Geometric Entities

Creating Named Selections

There are several ways to create named selections. Toolbar buttons are available on settings windows and The Selection List Window (see Table 6-2) to help group the geometric entities into manageable and easily identifiable selections that can be chosen from the **Selection** list on a settings window, for example.

- Create a Selection using the Definitions Node
- Create an Explicit Selection from a Settings Window
- Copying and Pasting Selection Lists
- Create an Explicit Selection from the Selection List Window

CREATE A SELECTION USING THE DEFINITIONS NODE

There are several types of selections that can be created. For each type, choose to add it from the **Definitions>Selections** submenu. In the settings window that opens, define the selection for that particular selection type. See The Types of Named Selections (Table 6-6) for a list and links to more information about the settings.

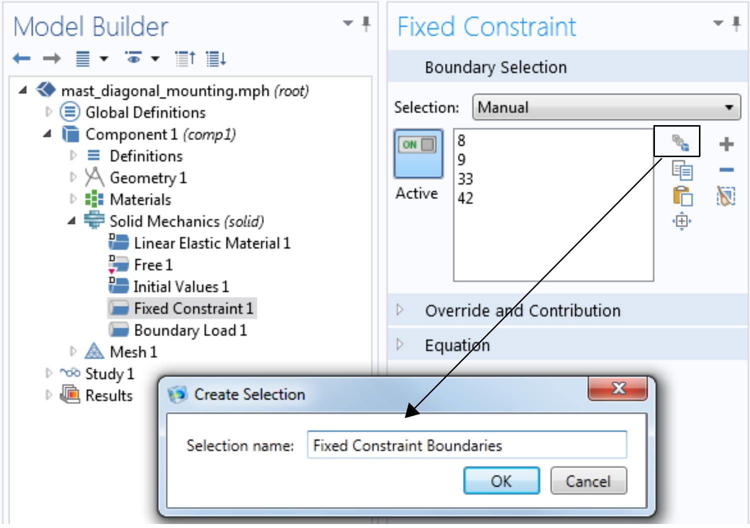
CREATE AN EXPLICIT SELECTION FROM A SETTINGS WINDOW

1 At any time during model creation, click a node that has the option to add a geometric entity to a selection, for example, under the **Materials** node or for the **Fixed Constraint** node for a **Solid Mechanics** physics interface as in Figure 6-14.

2 In the settings window that opens, select an option from the **Selection** list, for example, **Manual** or **All boundaries**.

3 Click the **Create Selection** button ( ) and enter a **Selection name** in the **Create Selection** window, for example, `Fixed Constraint Boundaries`. Click **OK** or press enter.

4 In the **Model Builder** the new **Explicit** node (now named `Fixed Constraint boundaries`) is added under **Definitions**. After creating these named selections, the **Selection** list displays the name in the list as in Figure 6-15.

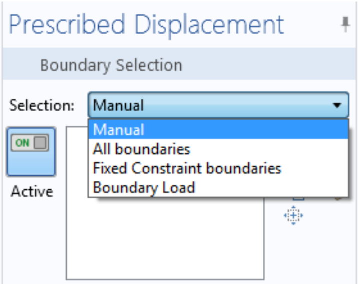


Selection: Manual

8
9
33
42

Selection name: Fixed Constraint Boundaries

Figure 6-14: Creating a selection from the settings window for a Fixed Constraint node.



Selection: Manual

Figure 6-15: After creating these named selections, the Selection list displays the new name in the list, making it simple to choose it when adding additional boundary conditions such as a Prescribed Displacement node.

CREATE AN EXPLICIT SELECTION FROM THE SELECTION LIST WINDOW

Continue using the same model for this example, which demonstrates how to identify specific boundaries to add to a new Explicit selection:

1 In the **Model Builder**, click the **Component 1** node.

2. Open The Selection List Window. Right-click and select **Float** to detach the window from the COMSOL Desktop.

> COMSOL displays the geometric entities in the **Selection List** based on where in the model you are working. Other nodes can be clicked to display the list of geometric entities. For example, the **Materials**, **Solid Mechanics** (in this example), **Mesh**, and **Geometry** nodes. Experiment in the COMSOL Desktop by clicking on different nodes and observing the changes in the **Selection List** and **Graphics** windows.

3. On the **Graphics** toolbar, click the **Select Boundaries** button ( ). The **Selection List** displays a list of all boundaries in the geometry. Click the **Select Domains**, **Select Edges**, or **Select Points** buttons in the **Graphics** window and observe how the list and the geometry changes based on the geometric entity level.
4. In the **Selection List** window, click to select the boundaries you want to add to an **Explicit** selection:
   - Click any individual boundary number in the list.
   - Shift-click to select contiguous items in the list.
   - Ctrl+click to select more than one boundary at a time.

   To help you identify the boundary, the selected boundaries are highlighted in red in the **Graphics** window.
5. Once the boundaries are chosen, click the **Create Selection** button ( ) and enter a **Selection name** in the **Create Selection** window, for example, `Tube boundaries`. Click **OK** or press enter.
6. Go to the **Model Builder**. The new **Explicit** node (now named `Tube boundaries`) is added under **Definitions**.

Copying and Pasting Selection Lists

Another way to create selections (see Creating Named Selections) is to copy and paste existing lists of geometric entities. If, for example, you have a list of geometric entities (boundaries, for example) in a file or document you can copy it to the clipboard and then use Ctrl+V to directly paste that list into a selection list in a settings window. An example of this is a list of geometric entity numbers described as a step in a modeling instruction.

You can also use the **Copy Selection** button ( ) and **Paste Selection** button ( ) if you have a list of geometric entities that you want to paste into a selection list. These buttons are available on many settings windows as in Figure 6-13 and Figure 6-14.

COPYING AND PASTING GEOMETRIC ENTITY INFORMATION INTO A SELECTION LIST

1. Prepare or copy the information to insert into the selection list. For example, copy a list of numbers from a text file or PDF file such as COMSOL model documentation (highlight and press Ctrl+C). Also copy a selection on any settings window, (for example, the Fixed Constraint boundaries). Click the **Copy Selection** button ( ) and go to the next step.
2. On the window or page next to the selection list where you want to add a selection from file (or on the clipboard), press Ctrl+V to paste the selection directly, or click the **Paste Selection** button ( ).
3. In the **Paste Selection** window, paste (press Ctrl+V) or enter the list of geometric entities into the **Selection** field. Data in the list or entered in the field can include commas and spaces as separators (1, 3), ranges (10–34), and words (*and*). Click **OK** to paste the selection into the selection list.

Adjacent

The **Adjacent** ( ) selection outputs selections adjacent to specified geometric entities or selections. For example, select all domains adjacent to some boundaries or all boundaries adjacent to some domains.

The adjacent geometric entities can be of any type (domains, boundaries, edges, or points) regardless of the geometric entity level for the input selections. To add this node, right-click the **Definitions** node and choose **Selections>Adjacent**.

INPUT ENTITIES

Based on space dimension, select a **Geometric entity level—Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Input selections** list and to create a selection of adjacent geometric entities. Click the **Add** button (+) to open an **Add** dialog box that contains selections of the chosen geometric entity level that appear earlier in the geometry sequence. Use the **Move Up** (↑), **Move Down** (↓), and **Delete** (✕) buttons to organize the list.

OUTPUT ENTITIES

In the **Geometry entity level** list, choose the type of output entities—**Adjacent domains**, **Adjacent boundaries**, **Adjacent edges** (3D only), or **Adjacent points**. If the output entities have a lower dimension that the input entities, there are also two check boxes that you can use to select exterior and interior entities of the union of the input selections.

By default, only exterior entities are selected. For example, if the input selections are domains selections, and the output is adjacent boundaries, the **Exterior boundaries** (selected by default) and **Interior boundaries** check boxes display.

- Creating Named Selections in the Geometry Sequence
- Adjacent Selection (Geometry Sequences)

Ball

Another way to select geometric entities is to define an enclosing **Ball** (●) to select geometric entities that are completely or partially inside the ball. To add this node, right-click the **Definitions** node and choose **Selections>Ball**.

GEOMETRIC ENTITY LEVEL

Select the **Level** for the geometric entities —**Domain**, **Boundary**, **Edge**, or **Point**.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions (to select all faces that make up a continuous sheet, for example).

When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).

See **Output Entities** for details about how the behavior depends on the condition for which the selection considers the group of entities to be enclosed.

INPUT ENTITIES

The **Entities** list defaults to **All**, which bases the selection on all entities of the selected type.

Select **From selections** to base the selection on other defined selections. Then, in the **Selections** list, add the selections for which you want to create a selection of geometric entities from those selections that are located within the ball, box, or cylinder that you define for the resulting selection. Click the **Add** button (+) to open an **Add** dialog box that contains selections of the chosen geometric entity level that appear earlier in the geometry sequence. Use the **Move Up** (↑), **Move Down** (↓), and **Delete** (✕) buttons to organize the list.

BALL CENTER/BALL RADIUS

Position the center of the ball by entering the center position in the **x**, **y**, and (3D only) **z** fields (the unit is the length unit for the geometry). Enter the radius for the ball (disk in 2D) in the **Radius** field. The default is 0.

OUTPUT ENTITIES

For the selections made under **Input Entities**, define the dimension of the ball and select the condition for the geometric entities to be selected. Choose an option from the **Include entity if** list—**Entity intersects ball** (the default), **Entity inside ball**, **Some vertex inside ball**, or **All vertices inside ball**.

- If **Entity intersects ball** is kept as the default, it includes all geometric entities that intersect the enclosing ball, that is, the selection includes all entities that are partially or completely inside the ball. If in addition the **Group by continuous tangent** check box is selected for a boundary or edge under the **Geometric Entity Level** section, all entities in each group are selected if any entity in the group intersects the ball.
- If **Entity inside ball** is chosen, it includes all geometric entities that are completely inside the enclosing ball. If in addition the **Group by continuous tangent** check box is selected for a boundary or edge under the **Geometric Entity Level** section, the entities in each group are selected only if all entities in the group are completely inside the ball.
- If **Some vertex inside ball** is chosen, it includes all geometric entities where at least some vertex is inside the enclosing ball. If in addition the **Group by continuous tangent** check box is selected for a boundary or edge under the **Geometric Entity Level** section, all entities in each group are selected if any entity in the group has at least some vertex inside the ball.
- If **All vertices inside ball** is chosen, it includes all geometric entities where all vertices are inside the enclosing ball. If the **Group by continuous tangent** check box is selected for a boundary or edge under the **Geometric Entity Level** section, the entities in each group are selected only if all entities in the group have all vertices inside the ball. This selection might differ slightly compared to when selecting **Entity inside ball** if the geometric entity is outside the ball at some points between the vertices.

The Ball, Box, and Cylinder selections use the rendering mesh to determine which entities fit the selection condition. You can control the detail for the rendering in the **Preferences** dialog box (select **Graphics** and then use the **Detail** list under **Visualization**).

- Creating Named Selections in the Geometry Sequence
- Ball Selection (Geometry Sequences)

Box

Another way to select geometric entities is to define an enclosing **Box** ( ) to select geometric entities that are completely or partially inside the box. To add this node, right-click the **Definitions** node and choose **Selections>Box**.

See Ball for the **Geometric Entity Level** and **Input Entities** settings.

BOX LIMITS

Define the dimensions of the box by entering the maximum and minimum values in all directions in the **x minimum**, **x maximum**, **y minimum**, **y maximum**, and (for 3D) **z minimum** and **z maximum** fields. The unit is the length unit for the geometry. The default is `–Inf` and `Inf` for the minimum and maximum values, respectively; that is, the box encloses the entire geometry. Use `–Inf` or `Inf` in some of these settings to make the box only partially bounded.

OUTPUT ENTITIES

For the selections made under **Input Entities**, define the dimension of the box and select the condition for the geometric entities to be selected. Choose an option from the **Include entity if** list—**Entity intersects box** (the default), **Entity inside box**, **Some vertex inside box**, or **All vertices inside box**.

See Ball for the settings. The only difference is that the settings are for a **Box** instead of a **Ball**.

- Creating Named Selections in the Geometry Sequence
- Box Selection (Geometry Sequences)

Cylinder

Another way to select geometric entities is to define an enclosing **Cylinder** ( ) to select geometric entities that are completely or partially inside the cylinder. To add this node, right-click the **Definitions** node and choose **Selections>Cylinder**.

See Ball for the **Geometric Entity Level** and **Input Entities** settings.

SIZE AND SHAPE

Define the dimensions of the cylinder by entering the radius and the positions of the upper and lower faces on the cylinder axis in the **Radius**, **Top distance**, and **Bottom distance** fields. The unit is the length unit for the geometry. The default is 0, `Inf`, and `-Inf` for these settings, respectively.

POSITION

Position the cylinder by entering the center position in the **x**, **y**, and (for 3D) **z** fields. The default is 0 for all coordinates.

AXIS

Set the cylinder axis by choosing an **Axis type**—**z-axis** (the default), **x-axis**, **y-axis**, **Cartesian**, or **Spherical**. If **Cartesian** is selected, enter coordinates for **x**, **y**, and **z**. If **Spherical** is selected, enter angles for **theta** and **phi** (unit: deg).

OUTPUT ENTITIES

For the selections made under **Input Entities**, define the dimension of the cylinder and select the condition for the geometric entities to be selected. Choose an option from the **Include entity if** list—**Entity intersects cylinder** (the default), **Entity inside cylinder**, **Some vertex inside cylinder**, or **All vertices inside cylinder**.

See Ball for the settings. The only difference is that the settings are for a **Cylinder** instead of a **Ball**.

- Creating Named Selections in the Geometry Sequence
- Ball Selection (Geometry Sequences)

Explicit

Use an **Explicit** node ( ) to create the selection using the selection tools for individual geometric entities (boundaries, for example) on the chosen geometric entity level. To add this node, right-click the **Definitions** node and choose **Selections>Explicit**.

INPUT ENTITIES

Based on space dimension, select a **Geometric entity level**—**Domain**, **Boundary**, **Edge**, or **Point** for the geometric entities to add or remove from the selection list.

Select and add geometric entities in the **Graphics** window, using other selection methods, or by selecting the **All domains**, **All boundaries**, **All edges**, or **All points** check box. The selected items are highlighted in the **Graphics** window. Selecting the check box for all geometric entities locks all entities of this type as selected even if the geometry changes.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions to select all faces that make up a continuous sheet, for example. When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).

To deselect one or some of the faces or edges that make up the group with continuous tangents, first clear the **Group by continuous tangent** check box.

OUTPUT ENTITIES

Define the geometry objects that the selection contains. The options available and defaults depend on the selection in the **Geometric entity level** list as well as the space dimension of the Component.

If **Domain** is the input the default output is the **Selected domains**. Select other options as required.

- Select **Adjacent boundaries**, **Adjacent edges**, or **Adjacent points** to use the boundaries, edges, or points next to the selected domains as the selection output (available options depend on the space dimension of the Component). This makes it possible to, for example, make a selection of all boundaries around a domain by first selecting the domain.
- Depending on the selection output, choose to include **Exterior boundaries** (the default) or **Interior boundaries**; **Exterior edges** (the default) or **Interior edges**; or **Exterior points** (the default) or **Interior points**. Click to select or clear the check boxes as needed.

If **Boundary** or **Edge** are chosen as the input, the default output is the **Selected boundaries** or **Selected edges**, respectively. As with the **Domain** input, select other options as required. The **Point** output is the same as the selection input (that is, the selected points).

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output type, for example, from domains to boundaries.

- Creating Named Selections in the Geometry Sequence
- Explicit Selection (Geometry Sequences)

Union, Intersection, Difference, and Complement

Boolean selections—**Union** ( ), **Intersection** ( ), **Difference** ( ), and **Complement** ( )—are useful to combine two or more selections (union), create a selection of overlapping geometric entities (intersection), create selection of entities that are in one selection but not in another (difference), and to create a selection as the complement (inverse) of one or more selections (all boundaries except the inflow boundaries, for example). The complement can be convenient if a selection is needed that consists of all boundaries except one or a few. Then it is easier to first create a selection (just the one boundary, for example) and then use a **Complement** node to create its complement. All Boolean selection nodes' settings windows have similar sections. To add this node, right-click the **Definitions** node and choose an option from the **Selections** menu.

GEOMETRIC ENTITY LEVEL

Based on space dimension, select a **Level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Selections to add** (**Selections to intersect**, **Selections to subtract**, **Selections to invert**) list.

INPUT ENTITIES

Use the buttons in this section to move, add, or delete selections in the **Selections to add** (Union), **Selections to intersect** (Intersection), or **Selections to invert** (Complement) lists. For the **Difference** selection node also choose **Selections to subtract**. Click the **Add** button (+) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Use the **Move Up** (↑), **Move Down** (↓), and **Delete** (☒) buttons to organize the list.

- Creating Named Selections in the Geometry Sequence
- Union Selection, Intersection Selection, Difference Selection, and Complement Selection (Geometry Sequences)

Creating Named Selections in the Geometry Sequence

You can also create selection nodes in the geometry sequence for user-defined named selections of all or a few of the geometric entities at a specific geometric entity level based on one or more of the nodes above the selection node in the geometry sequence. This way it is possible to make a selection that only includes a few of the geometric entities from one or more geometry objects and also create selections based on entire geometry objects. Using selections based on a geometry object makes it possible to track, for example, all boundaries in the final geometry that belong to that geometric object, even if its original boundaries are intersected by other geometry objects during a parametric sweep, for example. That is, the selection nodes in the **Geometry** branch can provide better associativity when changing or updating the geometry than the corresponding selection nodes under **Definitions**. See the following sections for details on the selection nodes in the geometry sequence, which you choose from the **Selections** submenu in the **Geometry** node's context menu.

- Working with Geometry Sequences
- Creating Named Selections

BUILDING SELECTION NODES

If the current node in the geometry sequence is before the node preceding the selection node, or after the selection node, the selection is not visualized (because the selected object or entities might not be visible in this state). In this case, the **Build Preceding State** button appears instead of the selection list (this also applies for nonexplicit selections in some selection nodes if **From selections** is selected under **Input Entities**). To visualize a state where the selection can be visualized, either click **Build Preceding State** or click the **Build Selected** button (▦). To rebuild the entire geometry, click the **Build All Objects** button (▦).

CREATING SELECTIONS FROM GEOMETRIC PRIMITIVES AND OPERATIONS

For all geometric primitives in 3D, 2D, and 1D geometries—for example, blocks, spheres, squares, polygons, and intervals—as well as for all geometry operations—for example, union, intersection, fillet, array, and mirror—you can create selections for each type of geometric entity that the resulting geometry objects consist of. The following list shows the geometric entity types for geometry objects that are “solids”:

- In 3D: domains, boundaries, edges, and points
- In 2D: domains, boundaries, and points
- In 1D: domains and points

To create these selections, click the geometry node in the **Model Builder** and then select the **Create selections** check box in the settings window’s **Selections of Resulting Entities** section. The selections become available with the name of the geometry node (Block 1, for example) in all applicable selection lists in the settings windows for the physics nodes, for example. There are no explicit Selection nodes for these selections. For example, for a 3D Component model with a single **Block** node, the **Selection** list contains the selection **Block 1**, which for a domain selection is the single domain, and for a boundary selection consists of the six faces of the block.

CUMULATIVE SELECTIONS

A *cumulative selection* is a selection in the geometry sequence that is a union of contributions from several selections. Cumulative selections are especially useful to construct a selection the has different definitions in different branches of an If clause. You can also use them as an alternative to a Union Selection node. There is no node in the model tree that corresponds to the cumulative selection.

For a selection geometry feature, you can let it contribute to an existing cumulative selection by choosing an existing cumulative selection in the **Contribute to** list in the **Cumulative Selection** section. To let it contribute to a new cumulative selection, click the **New** button. When a selection geometry feature contributes to a cumulative selection, the **Model Builder** displays the name of the cumulative selection instead of the original name of the selection geometry feature.

For a geometry feature that has a **Selection of Resulting Entities** section in its Settings window, you can similarly let the created selections contribute to a cumulative selection by choosing an existing cumulative selection in the **Contribute to** list, or by clicking the **New** button.

When a selection contributes to a cumulative selection, the original selection does not appear in lists where you choose among selections—instead, the cumulative selection replaces it. To remove a contribution to a cumulative selection, select **None** in the **Contribute to** list.

- Global Definitions and Geometry Subsequence
- Working with Geometry Sequences

TABLE 6-7: GEOMETRY NAMED SELECTIONS BY TYPE

ICON	TYPE	DESCRIPTION
	Adjacent Selection	Use this node to create the selection as the adjacent geometric entities (boundaries, for example) to one of more selections. See Adjacent Selection (Geometry Sequences).
	Ball Selection	Use this node to create the selection by enclosing part of the geometry by a bounding ball (sphere) to select geometric entities that are partially or completely inside the ball. See Ball Selection (Geometry Sequences).
	Box Selection	Use this node to create the selection by enclosing part of the geometry by a bounding box to select geometric entities that are partially or completely inside the box. See Box Selection (Geometry Sequences).

TABLE 6-7: GEOMETRY NAMED SELECTIONS BY TYPE

ICON	TYPE	DESCRIPTION
	Cylinder Selection	Use this node to create the selection by enclosing part of the geometry by a bounding cylinder to select geometric entities that are partially or completely inside the cylinder. See Cylinder Selection (Geometry Sequences).
	Explicit Selection	Use this node to create the selection using the normal selection tools for individual geometric entities (boundaries, for example) on the geometric entity level chosen. See Explicit Selection (Geometry Sequences).
Union Selection, Intersection Selection, Difference Selection, and Complement Selection (Geometry Sequences)		
	Union Selection	Use this node to create the selection as the union (addition) of two or more selections.
	Intersection Selection	Use this node to create the selection as the intersection of two or more selections.
	Difference Selection	Use this node to create the selection as the difference between a set of one or more selections and another set of one or more selections.
	Complement Selection	Use this node to create the selection as the complement (inverse) of one or more selections.

Adjacent Selection (Geometry Sequences)

The **Adjacent Selection** ( ) node selects all entities of a given dimension that are adjacent to entities in a given set of selections (having another dimension). For example, it can select all boundaries adjacent to a given domain selection. To add this node, right-click the **Geometry** node and choose **Selections>Adjacent Selection**.

See Adjacent for the **Input Entities** and **Output Entities** settings, which are the same.

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**. Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

- Named Selections
- Adjacent

Ball Selection (Geometry Sequences)

Another way to select geometry objects or geometric entities is to use an enclosing **Ball Selection** ( ) to select objects or entities that are completely or partially inside the ball. To add this node, right-click the **Geometry** node and choose **Selections>Ball Selection**.

See Ball for all settings. You can also choose **Object** as the **Level** under **Geometric Entity Level**.

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**.

Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

- Named Selections
- Ball, Box, and Cylinder

Box Selection (Geometry Sequences)

Another way to select geometry objects or geometric entities is to use an enclosing **Box Selection** ( ) to select objects or entities that are completely or partially inside the box. To add this node, right-click the **Geometry** node and choose **Selections>Box Selection**.

See Ball for the **Geometric Entity Level**, **Input Entities** and **Output Entities** settings. See Box for the **Box Limits** settings.

- You can also choose **Object** as the **Level** under **Geometric Entity Level**.
- For the **Output Entities** settings, note that the settings are for a **Box** instead of a **Ball**.

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**. Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

- Named Selections
- Ball, Box, and Cylinder

Cylinder Selection (Geometry Sequences)

Another way to select geometry objects or geometric entities is to use an enclosing **Cylinder Selection** ( ) to select objects or entities that are completely or partially inside the cylinder. To add this node, right-click the **Geometry** node and choose an option from the **Selections** menu.

See Ball for the **Geometric Entity Level**, **Input Entities** and **Output Entities** settings. See Cylinder for the **Size and Shape**, **Position**, and **Axis** settings.

- You can also choose **Object** as the **Level** under **Geometric Entity Level**.
- For the **Output Entities** settings, note that the settings are for a **Cylinder** instead of a **Ball**.

GEOMETRIC ENTITY LEVEL

Select the **Level** for the geometric entities —**Domain**, **Boundary**, **Edge**, or **Point**.

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**.

Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

- Named Selections
- Cylinder

Explicit Selection (Geometry Sequences)

Use an **Explicit Selection** node ( ) to create the selection using the selection tools for individual geometry objects or geometric entities (boundaries, for example). To add this node, right-click the **Geometry** node and choose **Selections>Explicit Selection**.

ENTITIES TO SELECT

Based on space dimension, select a **Geometric entity level**—**Object**, **Domain**, **Boundary**, **Edge**, or **Point** for the geometric objects or entities to add to the selection list.

Select and add geometric entities in the **Graphics** window or using other selection methods. The selected items are highlighted in the **Graphics** window.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions to select all faces that make up a continuous sheet, for example. When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the geometric entity level, for example, from domains to boundaries.

To deselect one or some of the faces or edges that make up the group with continuous tangents, first clear the **Group by continuous tangent** check box.

- Named Selections
- Explicit

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**. Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

Union Selection, Intersection Selection, Difference Selection, and Complement Selection (Geometry Sequences)

Boolean selections—**Union Selection** ( ), **Intersection Selection** ( ), **Difference Selection** ( ), and **Complement Selection** ( )—are useful to combine two or more selections (union), create a selection of overlapping geometric entities (intersection), create selection of entities that are in one selection but not in another (difference), and to create a selection as the complement (inverse) of one or more selections (all boundaries except the inflow boundaries, for example). The complement can be convenient if a selection is needed that consists of all boundaries except one or a few. Then it is easier to first create a selection (just the one boundary, for example) and then use a **Complement** node to create its complement. All Boolean selection nodes' settings windows have similar sections. To add this node, right-click the **Geometry** node and choose an option from the **Selections** menu.

GEOMETRIC ENTITY LEVEL

Based on space dimension, select a **Level**—**Object**, **Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Selections to add** (**Selections to intersect**, **Selections to subtract**, **Selections to invert**) list.

INPUT ENTITIES

Use the buttons in this section to move, add, or delete selections in the **Selections to add** (Union), **Selections to intersect** (Intersection), or **Selections to invert** (Complement) lists. For the **Difference** selection node also choose **Selections to subtract**. Click the **Add** button (+) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Use the **Move Up** (↑), **Move Down** (↓), and **Delete** ( ) buttons to organize the list.

CUMULATIVE SELECTION

See Cumulative Selections. From the **Contribute to** list, choose an option. If no previous **Cumulative Selection** has been created, **None** is available from the list. To populate the list and to create a new **Cumulative Selection**, click **New**. Enter a **Name** in the **New Cumulative Selection** window. Click **OK** or press enter. This is now available as an option from any **Contribute to** list for a Geometry selection.

- Named Selections
- Union, Intersection, Difference, and Complement

User-Defined Views

Views provide the camera setting, grid, rendering, arrows, lighting, and transparency in the **Graphics** window. You can create and use several user-defined views to highlight and display the geometry in different ways.

By default a **View** node is added to all 2D ( ), 2D axisymmetric ( ), and 3D ( ) models.

For 2D and 2D axisymmetric models, an Axis ( ) subnode is also added where you can set the axis coordinates and manual spacing.

The **View** nodes (View (2D)) and subnodes have information about a view. For 2D and 2D axisymmetric models, it controls the settings to display or hide geometry labels and direction arrows, and lock the axis.

For 3D models, a **Camera** subnode ( ) and three **Directional Light** nodes ( ), including default settings, are also added.

The **View** nodes (View (3D)) and their subnodes have information about a view. For 3D models the settings include (in addition to, for example, settings for displaying geometry labels and direction arrows) transparency, lighting sources, lighting attributes, and camera settings.

Also right-click the **View** node to add Hide Geometry Objects and Hide Geometric Entities nodes for any space dimension.

Right-click **Definitions** ( ) and select **View** to create additional **View** nodes and then experiment by switching views to find the best way to illustrate a model. Views can be selected from the list of views ( ) on the **Graphics** toolbar.

To reset the **View** node settings and its subnodes to the defaults, right-click **View** and select **Reset to Default** ( ).

To display the **Views** node under **Results** ( ), click the **Show** button ( ) and select **Advanced Results Options**. This is useful, for example, when 2D axisymmetric revolved plots or 2D cut plane plots for 3D models are created.

View (2D)

For 2D and 2D axisymmetric models, the **View** node ( ) controls the settings to display or hide geometry labels and lock the axis. An Axis node is added by default.

Also right-click the **View** node to add Hide Geometry Objects and Hide Geometric Entities nodes. To add additional **View** nodes, in the **Model Builder**, right-click **Definitions** and select **View**.

VIEW

Select the **Show geometry labels** check box to display the geometry object labels and the geometric entity labels (numbering) in the **Graphics** window. The labels appear for geometry objects or geometric entities (domain, boundary, edge, or point numbers), depending on what part of the model tree you display and the selection mode you are using.

Select the **Show edge direction arrows** check box to display the direction arrows on boundaries (edges) in the **Graphics** window. The direction arrows indicate the directions in which the boundary parameterization's value increases.

Select the **Lock axis** check box to store the current axis limits so that the zoom tools can be temporarily used, for example, but by revisiting the **View** node restore the axis limits to the values in the view at the time the **Lock axis** check box is selected.

View (3D)

Axis

For 2D and 2D axisymmetric models, the View (2D) node has an **Axis** ( ) subnode where the axis coordinates are set, manual spacing is defined, and the aspect ratio preserved.

AXIS

Enter **x minimum**, **x maximum**, **y minimum**, and **y maximum** values for the axis limits (**r** and **z** replace **x** and **y** in 2D axial symmetry). The **Preserve aspect ratio** check box is selected by default to make the increments equal in the *x*- and *y*-directions. Click to clear this check box to make the geometry fill the graphics window with unequal increments. This can be useful when working with slender geometries.

GRID

Select the **Manual spacing** check box and enter **x spacing** and **y spacing** values to control the grid spacing manually. **r** and **z** replace **x** and **y** in 2D axial symmetry. Enter **Extra x** and **Extra y** values directly or click the **Range** button ( ) as required.

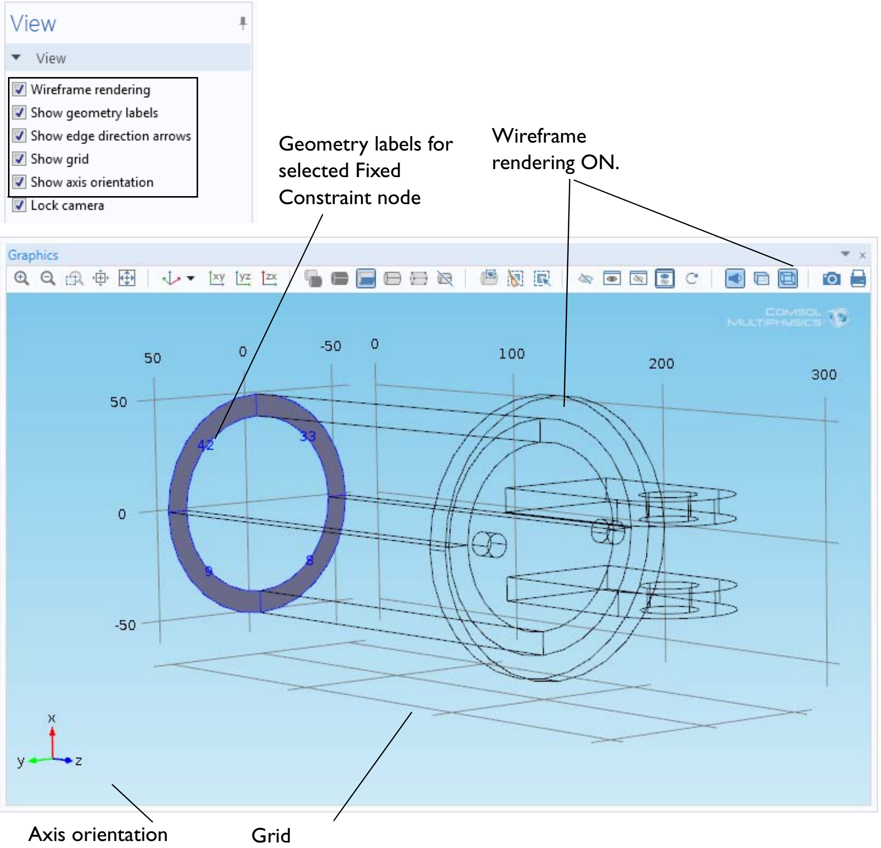
The default precision for the 2D grid axis labels is 4 digits. You can change the precision in the **Preferences** dialog box, using the **2D axis** field under **Precision** on the **Graphics** page.

View (3D)

View (3D)

The **View** node ( ) for 3D models has many options to add light sources and define the light attributes. Other functions include displaying or hiding geometry labels, transparency, wireframe rendering, a numbered grid, and axis orientation in the **Graphics** window. See Figure 6-16.

- View (2D)
- About the 3D View Light Sources and Attributes



View

View

☑ Wireframe rendering
☑ Show geometry labels
☑ Show edge direction arrows
☑ Show grid
☑ Show axis orientation
☑ Lock camera

Geometry labels for selected Fixed Constraint node

Wireframe rendering ON.

Axis orientation

Grid

Figure 6-16: An example of the top of 3D View settings window with all View option check boxes selected. The Graphics window shows what the check boxes represent. In this example, using the Diagonal Mounting Detail of a Communication Mast model, the Fixed Constraint node is clicked in the Model Builder, which then displays the numbers associated to the boundaries (8, 9, 33, and 42). Compare to the boundary numbers shown in Figure 6-14*, which is for the same node. The edge direction arrows are not displayed in this view.*

The **View** node () for 3D models has the following sections:

VIEW

- Select the **Wireframe rendering** check box to view the edges of the object as solid lines. The **Wireframe Rendering** button () is turned on or off in the **Graphics** window at the same time.
- Select the **Show geometry labels** check box to display the geometric object names and geometric entity labels (numbers) in the **Graphics** window. The labels appear for geometry objects or geometric entities (domain, boundary, edge, or point numbers), depending on what part of the model tree you display and the selection mode you are using.
- Select the **Show edge direction arrows** check box to display direction arrows on edges in the **Graphics** window. The direction arrows indicate the directions for which the edge parameterization values increase.
- By default the **Show grid** check box is selected and displays a numbered grid in the **Graphics** window around the object. Click to clear the check box to hide the grid.

- By default the **Show axis orientation** check box is selected and the axis orientation indicator for the global Cartesian coordinate directions is displayed in the lower-left corner of the **Graphics** window. Click to clear the check box to hide the axis orientation indicator.
- Select the **Lock camera** check box to store the current camera settings so that the zoom tools can temporarily be used, for example, but then revisiting the **View** node restores the camera settings to the values in the view at the time the **Lock camera** check box was selected.

LIGHT

The **Scene light** setting is a default that always displays and is based on the geometry. The **Scene light**, **Diffuse light**, **Specular light**, and **Ambient light** check boxes are selected by default. To hide and disable all light sources, click to clear the **Scene light** check box. The **Scene Light** button ( ) is turned on or off in the **Graphics** window at the same time.

- Click to clear the **Diffuse light**, **Specular light**, and **Ambient light** check boxes as required.
- Enter a value between 0 and 1 for the **Ambient intensity** (default value: 0.3) or use the slider to select a level. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color** from the list—**Custom**, **Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White** (default), or **Yellow**. The color is only applied to ambient light. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.

TRANSPARENCY

Select the **Transparency** check box to turn on transparency. The **Transparency** button ( ) is activated in the **Graphics** window at the same time. Enter a value between 0 and 1 in the **Alpha** field for the alpha blending, where 0 means a fully transparent color and 1 means a fully opaque color, or use the slider to select a transparency level. Watch the changes in the **Graphics** window at the same time to help choose a level.

Camera

Use the **Camera** node ( ) to orient the camera view in 3D models. In the **Model Builder**, under **Definitions** click to expand the **View** node where you want to define the camera position.

CAMERA

From the **Projection** list, select **Perspective** (the default) or **Orthographic** (parallel) as required. The perspective projection shows distant objects as smaller to provide a realistic 3D projection, whereas the orthographic projection shows all objects using their actual size regardless of the distance.

Enter a **Zoom angle** (in degrees) or use the **Zoom** buttons on the **Graphics** toolbar.

By default, the **Preserve aspect ratio** check box is selected to make the increments equal in the *x*-, *y*-, and *z*-directions. Clear this check box to make the geometry fill the graphics window with unequal increments. This can be useful when working with slender geometries.

POSITION

In the **Graphics** window, left-click and hold the mouse to orient the geometry on the axes, or enter **x, y**, and **z** coordinates.

TARGET

In the **Graphics** window, left-click and hold the mouse to orient the geometry on the axes. The corresponding coordinates are displayed in the settings window under the **Position**, **Target**, and **Up Vector** sections, or enter **x, y**, and **z** coordinates for the camera target location.

The **Position** is the location of the camera and the **Target** default is 1 length unit in front of the camera position.

UP VECTOR

In the **Graphics** window (or other plot windows), left-click and hold the mouse to orient the geometry on the axes, or enter **x, y**, and **z** coordinates for the camera's up vector, which determines which direction is up in the plot window.

CENTER OF ROTATION

By default the values in the **Center of Rotation** section define the center of the geometry in the view. Explicitly control the center of rotation by entering a center location in the **x**, **y**, and **z** fields and then click the **Apply** button (⟳) at the top of the settings window.

VIEW OFFSET

Right-click the mouse and move the geometry left, right, up, or down as required. This shift operation moves the currently visible frame on the image plane. The corresponding dimensionless values that display in the settings window under **View Offset** are relative to the image width and height, respectively. For example, an offset of $x = 0.5$ moves the projection 0.5 screen widths to the left. Alternatively, enter **x** and **y** values for the view offset.

The value in the **Orthographic scale** field defines the size in scene length of the viewing block along the longest side of the canvas. If the view's camera setting uses orthotropic projection, zoom in or out by increasing or decreasing the value of the orthographic scale. For a perspective projection this setting has no effect.

About the 3D View Light Sources and Attributes

Light sources are the **Directional Light**, **Point Light**, **Spotlight**, or **Headlight** nodes. *Light attributes* are the scene light components, which include diffuse, specular, and ambient light (see Figure 6-17). Combined, these attribute and source settings enable the software to render the 3D model to look realistic.

LIGHT ATTRIBUTES

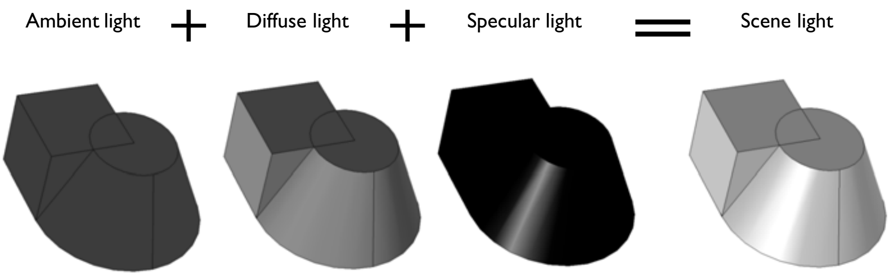


Figure 6-17: Scene light is a combination of ambient, diffuse, and specular light attributes. The default Scene light color is white and in this example the attributes display as different shades of black and gray. The Scene light is further enhanced using the various light source nodes.

Scene Light

For all geometry, scene light is applied by default and is a combination of the different light attributes. The light intensity and ambient intensity levels are also attributes of the scene light. The diffuse, specular, and ambient light attributes can be turned on or off by selecting or clearing the corresponding check boxes (see View (3D)). The intensity levels are adjusted either with a slider or by entering a number between 0 and 1.

Think of **Scene light** as being comprised of ambient light, the base amount of light, plus specular light to add depth to curves and diffuse light to soften the lighting and add contrast. See Figure 6-17 for examples of the attributes:

- **Ambient light** is the available light surrounding the geometry. By itself, ambient light makes a 3D object look like a 2D object. The addition of diffuse and specular light adds the contrast and depth needed to define 3D geometry.

- **Diffuse light** is directional and spreads out over the object, like a flashlight shining on a sphere. This generally adds contrast and depth of field to 3D objects.
- **Specular light** is directional and reflects off the surface of a sphere or curve in a geometry. It is based on the angle between the viewer and the light source.

Apply the different sources of light (**Directional Light**, **Point Light**, **Spotlight**, or **Headlight** nodes) to further enhance how the geometry displays.

LIGHT SOURCES

For 3D models, you can also add these light source nodes—**Directional Light** ( ), **Point Light** ( ), **Spotlight** ( ), and **Headlight** ( )—to adjust how the color and shading displays in the Graphics window. Each View can have a maximum of eight light sources (nodes) in any combination. See Directional Light, Point Light, Spotlight, and Headlight below.

Each light source has a unique light marker displayed in the Graphics window. Figure 6-19 shows three directional light settings and markers displayed in the Graphics window. The light markers are placed at the user-defined *x*, *y*, and *z* coordinates and are used to adjust the light and specular intensities on the object. You can show or hide the markers and change the color. The color of the marker corresponds to the light hitting the object. A wireframe around a light marker indicates that its node is selected in the Model Builder. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object. When adjusting the spread angle for a spotlight, the arrow increases and decreases in width as the angle value increases and decreases.

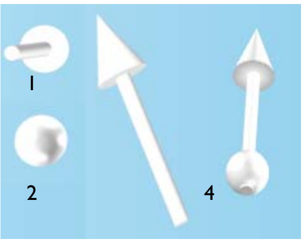


Figure 6-18: The light markers for each type of light source: (1) Headlight,(2) Point light, (3) Direction light, and (4) Spotlight.

Directional Light

By default, a **View** contains three **Directional Light** ( ) nodes. Directional light represents light that falls from a direction on all objects in the scene, like sunlight where all the light rays are parallel. Directional lights therefore have no position. The direction of the light, the light and specular intensity levels, and the color can be adjusted as required.

Figure 6-19 is an example of three **Directional Light** nodes where the color is changed and the light intensity adjusted for Directional Light 3. The markers are labeled 1, 2, and 3 to correspond to the nodes. The **Directional Light 3** node's light intensity setting is changed from 0.24 to 1—the change in arrow size corresponds to the increase in light intensity. The wireframe around a light marker means that the corresponding node is selected in the **Model Builder**. Adjust the direction of the light, the light and specular intensity levels, and the color as required.

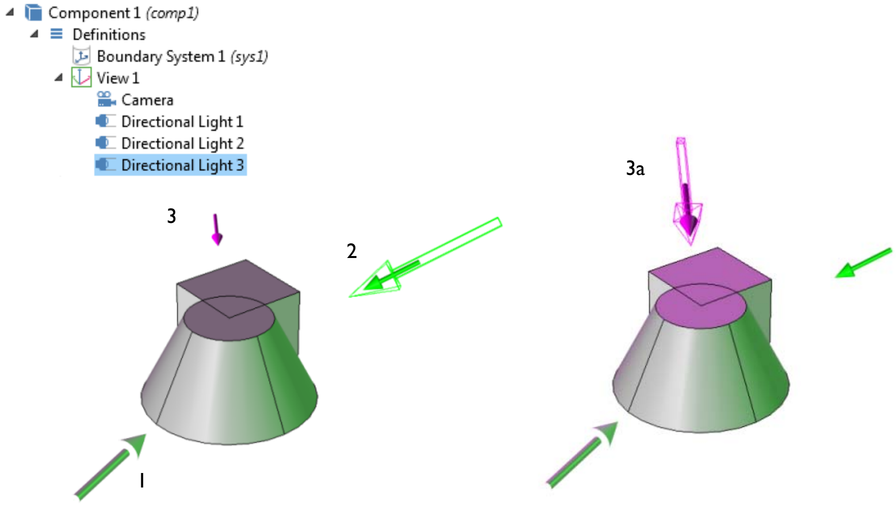


Figure 6-19: Examples of directional light markers and the location of each directional light. The markers and light can be color coded to see which areas need adjustment. The markers indicate which node is selected in the Model Builder (a wireframe around the arrow in 2 and 3a), and the level of light intensity applied (the size of the arrow; compare 3 and 3a).

DIRECTION

The **x**, **y**, and **z** coordinates define the direction in which the light falls on the objects in the scene. The arrow is a visualization of that direction with a placement that is calculated automatically depending of the scene's bounding box. The arrow points in the direction of the light.

Change the color to something other than white to observe where the light hits the object.

SETTINGS

Adjust some of the settings such as the intensity and color of the directional light.

- Enter a **Light Intensity** or use the slider to select values. Watch the changes in the **Graphics** window at the same time to help choose a level. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object.
- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and used to define spheres and curves). This setting is turned ON and OFF on the View page (the default is ON), and the levels are adjusted for each **Specular intensity** using the field or the slider to select values between 0 and 1 (default value: 1). Watch the changes in the **Graphics** window at the same time to help choose a level. See Figure 6-17 for an example of specular light.
- Select a **Color—White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.

- By default, the **Lock to camera coordinate system** check box is selected to make the light rotate together with the camera. Click to clear as required; the light then follows the geometry.

To lock the camera settings, see Camera.

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See Figure 6-18 for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Point Light

The **Point Light** (💡) has a position and emits the light equally in all directions; it is like a light bulb. Point light have therefore no direction. The position of the light, the light and specular intensity levels, and the color can be adjusted as required. Figure 6-20 is an example of two **Point Light** nodes added to the **View** and all other nodes disabled.

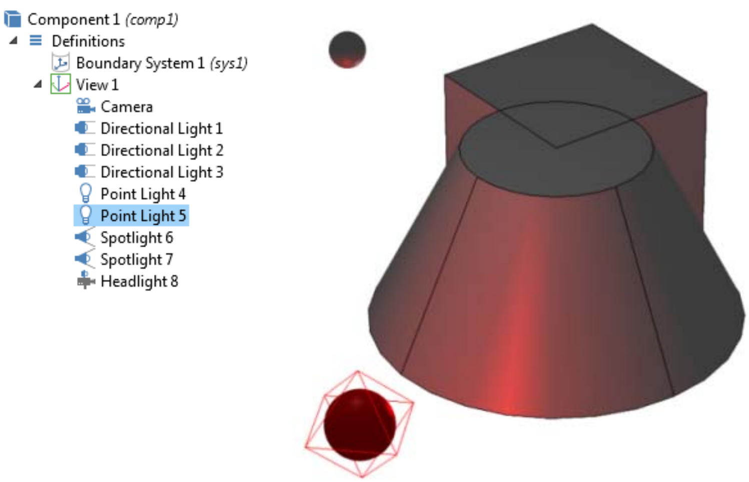


Figure 6-20: An example of two Point Light nodes added to a View. All other light nodes are disabled to show only the Point Light effects to the geometry. Point Light 5 is selected in the Model Builder as indicated by the wireframe around the light marker.

POSITION

Enter **x**, **y**, and **z** coordinates for the point light's position.

SETTINGS

Adjust some of the settings such as the intensity and color of the point light.

- Enter a **Light Intensity** or use the slider to select values. Watch the changes in the **Graphics** window at the same time to help choose a level. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object.
- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and used to define spheres and curves). This setting is turned on and off on the View page (the default is on), and the levels are adjusted for each **Specular intensity** using the field or the slider to select values between 0 and 1 (default value: 1). Watch the changes in the **Graphics** window at the same time to help choose a level. See Figure 6-17 for an

example of specular light.

- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
- By default, the **Lock to camera coordinate system** check box is selected to make the light rotate together with the camera. Click to clear as required; the light then follows the geometry..

To lock the camera settings, see Camera.

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See Figure 6-18 for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Spotlight

The **Spotlight** ( ) acts like a flashlight and has both a position and a direction. Figure 6-21 is an example of two **Spotlight** nodes added to a **View** with all other nodes disabled. The position and direction of the light, the light and specular intensity levels, and the color can be adjusted as required. In addition, the spread angle can be adjusted as shown in the figure. The width of the light marker corresponds to the spread angle.

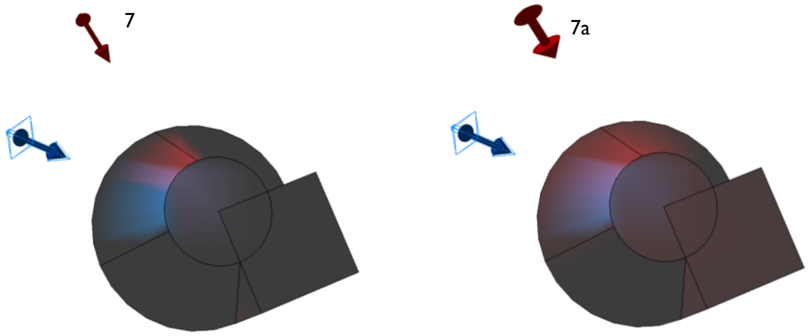


Figure 6-21: Two Spotlight nodes displayed with all other nodes disabled showing where the Spotlight is focused. When the spread angle is changed from 20 to 100 for Spotlight 7 (as indicated by the change in marker width between 7 and 7a), a corresponding change is made to the model.

POSITION

Enter **x**, **y**, and **z** coordinates for the position of the spotlight.

DIRECTION

The **x**, **y**, and **z** coordinates define the direction in which the spotlight falls on the objects in the scene. The arrow points in the direction of the light.

Change the color to something other than white to observe where the light hits the object.

SETTINGS

Adjust some of the settings such as the spread, intensity, and color of the spotlight.

- Enter a **Spread angle** (in degrees). Also watch the changes in the **Graphics** window at the same time to help choose an angle. When adjusting the spread angle, the arrow increases and decreases in width as the angle value increases and decreases. The default is 20 degrees. See Figure 6-21 for an example.
- Enter a **Light Intensity** or use the slider to select values. Watch the changes in the **Graphics** window at the same time to help choose a level. The light marker's length changes as the corresponding light intensity changes.
- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and used to define spheres and curves). You can turn specular light on and off in the **View** node's settings window (default: on), and the levels are adjusted for each **Specular intensity** using the field or the slider to select values between 0 and 1 (default value: 1). Watch the changes in the **Graphics** window at the same time to help choose a level. See Figure 6-17 for an example of specular light.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
- By default, the **Lock to camera coordinate system** check box is selected to make the light rotate together with the camera. Click to clear as required; the light then follows the geometry.

To lock the camera settings, see Camera.

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See Figure 6-18 for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Headlight

A headlight is a directional light that points to the scene from the camera position. The **Headlight** ( ) is similar to the **Directional Light** with the only difference being that it is always locked to the camera's coordinate system and a direction pointing from the camera is used. Figure 6-22 is an example of one **Headlight** node added to a **View** with the **Directional Light** nodes enabled. The headlight's position and direction cannot be changed; it is based on the **Directional Light** node's (or nodes') x, y, and z coordinates. The light and specular intensity levels and the color can be adjusted as required.

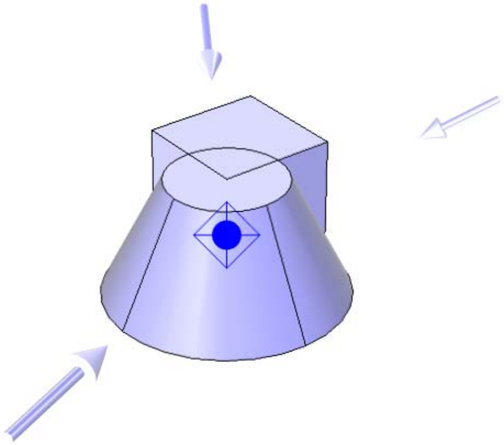


Figure 6-22: An example of a blue Headlight source with three Directional light sources set to white. If the geometry is rotated, you can adjust and view the effects of the Headlight source on the geometry based on the shades of blue and white.

SETTINGS

Adjust the settings such as the intensity and color of the headlight.

- Enter a **Light Intensity** or use the slider to select values. Watch the changes in the **Graphics** window at the same time to help choose a level. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object.
- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and used to define spheres and curves). This setting is turned ON and OFF on the View page (the default is ON), and the levels are adjusted for each **Specular intensity** using the field or the slider to select values between 0 and 1 (default value: 1). Watch the changes in the **Graphics** window at the same time to help choose a level. See Figure 6-17 for an example of specular light.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See Figure 6-18 for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Hide Geometry Objects and Hide Geometric Entities

In the **Model Builder**, under **Definitions**, right-click any **View** node and select **Hide Geometry Objects** (to hide a set of geometry objects or some of their geometric entities—in the created or imported geometry. Right-click **View** and select **Hide Geometric Entities** ( ) to hide geometric entities (boundaries, for example)—in the analyzed (finalized) geometry used for modeling—at a specific geometric entity level for any View.

> The **Hide Geometry Objects** node hides geometry objects or parts of such objects and affects the view in the Geometry branch only. The **Hide Geometric Entities** node hides geometric entities (domains, boundaries, edges, or points) in the finalized geometry and affects the view in the Materials, physics, Mesh, studies, and Results branches.

SELECTION

Hide a set of geometry objects at a specific **Geometric entity level**—**Object** (the default), **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Then select the geometry objects or some of their geometric entities in the **Graphics** window. The list under the **Geometric entity level** list contains the selected objects or entities to hide in the **Geometry** branch. For example, for **Hide Geometry Objects**, hide a geometry objects, or hide a boundary of a 3D geometry object to make it possible to view the inside of the object. Or, for **Hide Geometric Entities**, hide some domains that are not of interest in the model, or some boundaries only on a specific domain to get a clearer view of some of the model. You can include the hidden objects (entities) in a plot by selecting the **Show hidden objects** check box under **Plot Settings** in the corresponding plot group's settings window.

The **Hide Geometric Entities** settings window contains the following section:

GEOMETRIC ENTITY SELECTION

Hide a set of geometric entities at a specific **Geometric entity level**—**Domain** (the default), **Boundary**, **Edge** (3D only), or **Point**. From the **Selection** list, select **Manual** (the default) or **All domains**, **All boundaries**, **All edges** (3D only), or **All points**. If **Manual** is selected, go to the **Graphics** window and select the geometric entities that you want to hide. The selected entities appear in the list under the **Selection** list. Use the **Add to Selection** (+), **Remove from Selection** (−), **Clear Selection** (🧹), and **Zoom Selected** (⊞) buttons as required.

7

Geometry Modeling and CAD Tools

The CAD tools in COMSOL Multiphysics® include many geometric primitives and operations for modeling the geometry using solid modeling and boundary modeling. This chapter covers geometry modeling in 1D, 2D, and 3D with examples of solid modeling, boundary modeling, Boolean operations, and other CAD tools. In addition, it shows how to use the tools for exploring geometric properties, such as volumes and surfaces. There is also information about using external CAD data.

Creating a Geometry for Analysis

Overview of Geometry Modeling Concepts

In COMSOL Multiphysics you can use *solid modeling* or *boundary modeling* to create objects in 1D, 2D, and 3D. These can be combined in the same geometry (*hybrid modeling*).

- During solid modeling, a geometry is formed as a combination of solid objects using *Boolean operations* like union, intersection, and difference. Objects formed by combining a collection of existing solids using Boolean operations are known as *composite solid objects*.
- Boundary modeling is the process of defining a solid in terms of its boundaries—for example, using lines to create a solid hexagonal domain in 2D. You can combine such a solid with *geometric primitives*—common solid modeling shapes like blocks, cones, spheres, rectangles, and circles, which are directly available in COMSOL.

In 3D, you can form 3D solid objects by defining 2D solids in *work planes* and then *extrude* and *revolve* these into 3D solids. It is also possible to *embed* 2D objects into the 3D geometry.

You can also overlay additional nonsolid objects on top of solid objects to control the distribution of the mesh and to improve analysis capabilities. For example, you can add a curve object to a geometry to control the element size in the vicinity of this curve, or add a point to guarantee a mesh vertex in a specific location or to create a time-dependent or parametric-value graph at that location in the geometry.

The settings for the nodes making up a geometry sequence can be changed at any time and the whole sequence can be re-run. It is also possible to parameterize the geometry using one or more parameters that define properties of a geometric primitive, for example. COMSOL then takes the parameterization into account as part of the geometry sequence for each step in a parametric sweep. You can also insert geometry sequences from other models into your current sequence.

You can import 2D geometries from DXF files and 3D geometries from STL and VRML files.

See Import for details of how to import these CAD file formats. Combining several objects where at least one is imported from an STL/VRML file is not supported.

The CAD Import Module provides an interface for the import of CAD files in Parasolid, SAT (ACIS), Inventor, Pro/E, SolidWorks, STEP, and IGES formats. In addition, the CATIA V5 Import Module provides an interface for CATIA V5 files.

The optional LiveLink™ products offer bidirectional links to 3D CAD software. Using these, you can run parametric geometry sweeps driven from the COMSOL environment but operating directly on the geometries in the respective CAD package.

- Working with Geometry Sequences
- Creating a Geometry Sequence
- Geometry in the *COMSOL API Reference Manual*

The *Introduction to COMSOL Multiphysics* includes a tutorial to learn how to build the busbar geometry. See Appendix A—Building a Geometry or the printed copy included with COMSOL Multiphysics.

Techniques for Creating Geometries

Several techniques can ensure that a geometry results in a good mesh and gives reasonable solution times for the analysis. They include the use of symmetry and eliminating small details, gaps, holes, and singularities.

USING SYMMETRIES

Using symmetry is one of the most effective ways to reduce the size of a model. For axially symmetric geometries, a 2D axisymmetric model is sufficient. You can easily visualize the results in a full 3D geometry using a Revolution 2D data set. Other common cases of symmetry are sector symmetry and symmetry and antisymmetry planes, which can reduce the size of a 3D model.

MAKING THE GEOMETRY MATCH THE BOUNDARY CONDITIONS

Sometimes the modeling domain is unbounded or too large for successful analysis. For those cases a suitable boundary condition can replace the exterior of the domain.

!	It is important that the geometry is large enough to validate the boundary conditions.

For outflows in fluid-flow models, for example, the boundary should be perpendicular to the fully developed flow. Inspections and modifications of the solved model might be necessary to verify the validity of the boundary condition. Also, for some applications, infinite elements or perfectly matched layers (PMLs) are available for modeling diffusion or wave propagation in unbounded domains.

AVOIDING EXCESSIVELY SMALL DETAILS, HOLES, AND GAPS

Many geometries, especially those designed using a CAD system, contain small holes, details, and gaps. These small features can make the domain unbounded and must be removed before analysis. Small details and holes can lead to large meshes or even failure during mesh generation. Make sure the snapping feature is activated to avoid small gaps and mismatches between the geometry objects.

The CAD Import Module contains tools for automatic and interactive repair and defeaturing of 3D CAD data. For a 2D or 3D model you can also remove small details and prepare the geometry for efficient meshing using virtual geometry operations (see Virtual Geometry Operations).

AVOIDING SINGULARITIES AND DEGENERACIES IN THE GEOMETRY

A singularity in a geometry is a sharp corner or angle that can create problems during meshing and analysis. In reality, a sharp reentrant corner leads to infinite stress values in a stress analysis of a perfectly elastic material. The stress value for a sharp corner is finite in the stress analysis, but refinement of the mesh increases the stresses in the corner without limit. To avoid a singularity, round sharp corners using fillets.

A degeneracy in the geometry can occur during solid modeling. For example, fillet areas that taper to a point and the apex of a cone can become degenerate points. These degeneracies might cause problems for the mesh generator and during the analysis. A common degeneracy in the geometry occurs when a 3D solid is created (for example, a cylinder) by rotation about an axis that touches the rotation area. It is then better to create the solid object by extruding a cross section or to use geometric 3D primitives.

Associative Geometry and Selections of Geometry Objects

Associative geometry is a concept for the automatic updating of applied physical properties, such as boundary conditions and equation coefficients, under geometric transformations. Thus, once you have defined the physical properties of a model and return to the Geometry branch to modify the geometric model, COMSOL Multiphysics updates the physical properties according to the geometry modifications. The associative geometry functionality utilizes geometry-mapping information between the groups of geometric entities (points, edges, boundaries, and domains) in the *finalized geometry* and the corresponding groups in the geometric model.

This geometry mapping is not always without ambiguities. COMSOL makes some heuristic decisions when mapping the physical properties between the finalized geometry (the object on which the physical properties are imposed) and the geometric model. In some cases the resulting updated physical properties might not be the ones that are expected.

User-defined named *selection nodes* in the geometry sequence are useful to improve associativity compared to other selection nodes (see Creating Named Selections in the Geometry Sequence). You can refer to such selections defined in following geometry nodes (for example, as input objects). This applies both to selections created by the **Create selections** check box and selections created by selection nodes.

Choosing the Right Space Dimension

Most of the problems solved with COMSOL Multiphysics are three-dimensional (3D) in the real world. In many cases, it is sufficient to solve a two-dimensional (2D) or one-dimensional (1D) problem that is close, or equivalent, to the real problem. 2D models are easier to modify and generally solve much faster, so modeling mistakes are much easier to find when working in 2D. Once the 2D model is verified, you are in a better position to build a 3D model.

Not all physics are available in all space dimensions. See the documentation for the physics interfaces in COMSOL and its modules for the supported space dimensions.

1D PROBLEMS

The following is a guide for some of the common approximations made for 1D problems. Remember that modeling in 1D usually represents some 2D or 3D geometry under the assumption that nothing changes in the other dimensions.

Cartesian Coordinates

In a 1D model you view a single straight line that represents the only space dimension where there is spatial (or other) variation.

Axial Symmetry (Cylindrical Coordinates)

In an axially symmetric 1D model you view a straight line that represents the radial direction in an axially symmetric geometry.

2D PROBLEMS

The following is a guide for some of the common approximations made for 2D problems. Modeling in 2D often represents a 3D geometry under the assumption that nothing changes in the third dimension.

Cartesian Coordinate Systems

In this case you view a cross section in the *xy*-plane of the actual 3D geometry. The geometry is mathematically extended to infinity in both directions along the *z*-axis, assuming no variation along that axis. All the total flows in and out of boundaries are per unit length along the *z*-axis. A simplified way of looking at this is to assume that the geometry is extruded one unit length from the cross section along the *z*-axis. The total flow out of each boundary is then from the face created by the extruded boundary (a boundary in 2D is a line).

There are usually two approaches that lead to a 2D cross-sectional view of a problem:

- When there is no variation of the solution in one particular dimension.
- When there is a problem where the influence of the finite extension in the third dimension can be neglected.

In some applications there are special 2D assumptions, such as the *plane strain* and *plane stress* conditions for 2D stress analysis in solid mechanics.

In addition to the unit-depth assumption, some physics interfaces (for solid mechanics and heat transfer, for example) provide the thickness as a user-defined property in 2D models. For heat transfer, the thickness is used when including out-of-plane heat transfer in the model.

Axial Symmetry (Cylindrical Coordinates)

If the 3D geometry can be constructed by revolving a cross section about an axis, and no variations in any variable occur when going around the axis of revolution, an axisymmetric physics interface can be used.

The spatial coordinates are called r and z, where r is the radius. The flow at the boundaries is given per unit length along the third dimension. Because this dimension is a revolution, you have to multiply all flows with αr, where α is the revolution angle (for example, 2π for a full turn). COMSOL provides this is an option during postprocessing.

Geometric Variables and Mesh Variables

3D PROBLEMS

This section discusses 3D geometry modeling practices.

Although COMSOL fully supports arbitrary 3D geometries, it is important to simplify the problem. This is because 3D problems easily get large and require more computer power, memory, and time to solve. The extra time spent on simplifying a problem is probably well spent when solving it.

Is it possible to solve the problem in 2D? Given that the necessary approximations are small, the solution is more accurate in 2D because a much denser mesh can be used. See 2D Problems if this is applicable.

Are there symmetries in the geometry and model? Many problems have planes where the solution on either side of the plane looks the same. A good way to check this is to flip the geometry around the plane, for example, by turning it upside down around the horizontal plane. You can then remove the geometry below the plane if you do not see any differences between the two cases regarding geometry, materials, and sources. Boundaries created by the cross section between the geometry and this plane need a symmetry boundary condition, which is available in all 3D physics.

Do you know the dependence in one direction so it can be replaced by an analytical function? You can use this approach either to convert 3D to 2D or to convert a layer to a boundary condition.

THE COORDINATE SYSTEMS AND THE SPACE DIMENSION

COMSOL uses a global Cartesian or cylindrical (axisymmetric) coordinate system. You select the geometry dimension and coordinate system when creating a new model. The default variable names for the spatial coordinates are x, y, and z for Cartesian coordinates and r, φ, and z for cylindrical coordinates. These coordinate variables (together with the variable t for the time in time-dependent models) make up the *independent variables* in COMSOL models.

The labels assigned to the coordinate system variables vary according to the space dimension:

- Models that are opened using the space dimensions 1D, 2D, and 3D use the Cartesian coordinate independent variable labels x, y (2D and 3D), and z (3D).
- In 2D axisymmetric geometries, the x-axis represents the r label, which is the radial coordinate, while the y-axis represents the z label, the height coordinate.
- In 1D axisymmetric geometries, the default radial coordinate is labeled r, and represented by the x-axis.

For axisymmetric cases the geometry model must fall in the positive half plane ($r \geq 0$).

- Creating a New Model
- Coordinate Systems
- Cylindrical System

The nodes under a physics in the Model Builder and Plot Groups and Plots use different icons and colors to help visualize a model.

- Physics Nodes by Space Dimension
- Plot Groups and Plots

Removing Interior Boundaries

Removing interior boundaries is good practice if the interior boundary is an effect of the geometry modeling and does not represent a border between different materials or between domains with different properties. When you remove the interior boundaries, the resulting geometry consists of fewer domains and puts fewer constraints on the mesh generation.

To remove interior boundaries, clear the **Keep interior boundaries** check box in a Boolean operations such as Union or Compose.

It is sometimes useful to keep interior boundaries for controlling the mesh. In such cases, use virtual operations such as Ignore Edges and Ignore Faces but keep the original interior boundaries for mesh control. The interior boundaries are then not part of the geometry for defining physics but are present during meshing to define areas where you want to use a finer mesh, for example. See Mesh Control Entities for more information.

Working with Geometry Sequences

The Geometry Nodes

Once you have added a Component, The Geometry Node, representing the *geometry sequence* of the model component, is added under the Component node. Initially, a geometry sequence only contains a **Form Union** node. The Component's geometry is created by adding nodes to the sequence and building them.

- To add features to a geometry sequence, use the buttons in The Geometry Toolbar or right-click The Geometry Node in the **Model Builder** and then select one of the available options. Then see Geometric Primitives, Geometry Operations, and Virtual Geometry and Mesh Control Operations for descriptions of the geometry features. The tables in each section link to the individual feature settings window descriptions.
- Learn about Creating a Geometry Sequence where you can use the buttons on the toolbars, right-click the Geometry node to add items from the context menu, copy and paste geometry features, insert an existing geometry sequence from another MPH file, or import a geometry. Sometimes Exporting a Geometry is useful.
- Once a geometry sequence is in place, you need to understand about Editing and Building Geometry Nodes, Measuring Geometry Objects, and The Form Union/Assembly Node—Uniting the Geometry.

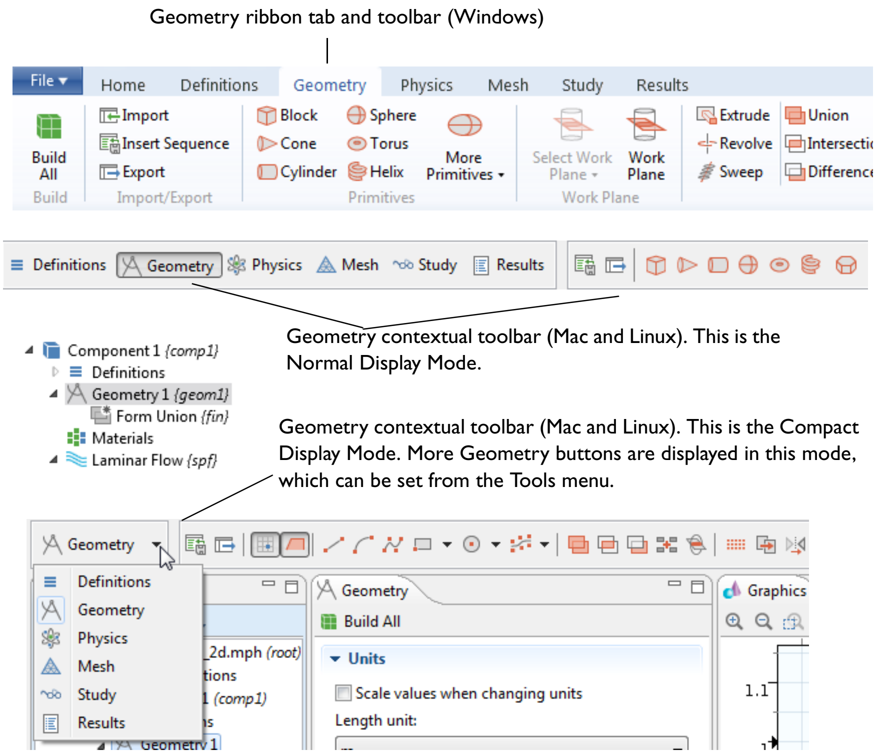
- Creating a New Model
- The Component Node
- Working with Nodes in the Model Builder
- Global Definitions and Geometry Subsequence
- Creating Named Selections in the Geometry Sequence

If you add Work Plane nodes in a 3D geometry, a Plane Geometry Node appears under the Work Plane node, and you can add 2D geometry objects and features under that node to create the 2D geometry sequence that defines the work plane's geometry.

The Geometry Toolbar

After a **Component** node is added to the model, the **Geometry** ribbon toolbar (Windows) or the **Geometry** contextual toolbar (Mac and Linux) is made accessible (Geometry Toolbar). Click the **Geometry** toolbar to display the options.

Geometric Primitives and Geometry Operations for example, can be added using these toolbars. You can also right-click The Geometry Node in the **Model Builder** and select any one of the available options. You can also use some drawing tools as listed in Geometry Drawing Toolbar Buttons.



Geometry ribbon tab and toolbar (Windows)

Geometry contextual toolbar (Mac and Linux). This is the Normal Display Mode.

Geometry contextual toolbar (Mac and Linux). This is the Compact Display Mode. More Geometry buttons are displayed in this mode, which can be set from the Tools menu.

Figure 7-1: Click the Geometry ribbon toolbar (Windows users, top) or Geometry contextual toolbar (Mac and Linux users, bottom) to display the options of the Geometry toolbar. This example is the 3D toolbar. Only some of the available buttons are shown in this image, and some features are not available yet, which is indicated by a grayed out button or icon. The contextual toolbar for cross platform users can also have a Normal or Compact display mode. This changes the number of geometry buttons available.

Options become available based on where in the geometry you are working. As in Figure 7-1, the **Geometry** toolbar has some options grayed out because these are not yet available. As a geometry sequence is built, and you select geometry objects in the **Graphics** window, the applicable options become available.

When one of the buttons is clicked, COMSOL performs the associated operation on the selected objects and creates the resulting objects, often adding a node to the geometry sequence. If you want to modify the operation, you can edit and rebuild this node. See About Highlighted Geometric Entities in the Graphics Window for details.

DRAWING GEOMETRIC PRIMITIVES

In 2D (and 1D) there are buttons for drawing Geometric Primitives by using the mouse. In 3D, the buttons are available to create primitives, but you cannot draw these using the mouse. See Geometry Toolbar and Table 2-8 for a list of the buttons and links to the individual features.

MOVING AND SCALING 2D OBJECTS IN THE GRAPHICS WINDOW

For 2D geometries, you can click the **Edit** button (✎) in the **Graphics** window's toolbar to enter an Edit selection mode, where you can perform the following actions:

- Click to select; then click-and-drag to move a geometry object in the 2D geometry plane.
- Click to select; then click- and-drag using one of the small squares in the frame that surrounds the selected objects to scale it. If the scaling is isotropic so that the geometry object retains its original shape and type, the geometry object's settings show the resulting size and position. If the scaling is anisotropic (stretching a square

so that it becomes a rectangle, for example), a **Scale** node appears under the geometry objects' nodes in the geometry sequence, and it contains the corresponding anisotropic scale factors.

- Alt-click to enter a mode where you can edit the geometry object by clicking-and-dragging control points in the Graphics window. Right-click to exit this mode; an **Edit Object** node then appears under the geometry objects' nodes in the geometry sequence, and it contains the properties of the edited object. See Editing 2D Geometry Objects below.

When you work in the Edit selection mode, there is no highlighting of objects when you hover over them with the mouse. Click to select a single geometry object; ctrl-click to select multiple objects. Click outside of the geometry object to deselect.

EDITING 2D GEOMETRY OBJECTS

Use the **Edit Object** (✏) node to adjust the edges and vertices for a 2D geometry object or to add or delete edges and vertices in the object. In the **Model Builder**, right-click a 2D **Geometry** and select **Edit Object**.

The **Edit Object** function can also be started using the **Edit Object** button (✏) in the Geometry toolbar:

1. In the **Model Builder** click the **Geometry** node.
2. Select a single geometry object in the **Graphics** window.
3. Click the **Edit Object** button (✏). The **Graphics** window displays handles for vertices and edge control points in the selected object.
4. Use the mouse to drag vertices and control points to new locations. The image is updated in the **Graphics** window to show the effect on the object being edited.
5. Right-click to exit the object editing mode and save the changes. A new **Edit Object** node is added in the **Model Builder**. Alternatively, if the object being edited is an **Edit Object** feature, the changes are incorporated in the existing feature and no new node is added in the **Model Builder**.
6. Click the left mouse button outside the object being edited to cancel the editing operation.

Another way to edit a geometry object, if you are in the Edit selection mode, is to hold down the Alt key and left-click on the object in the **Graphics** window. This is equivalent to selecting the object and then using the **Edit Object** toolbar button.

See Edit Object for details.

The Geometry Node

Under a **Geometry** node ( ) you define and create the geometry sequence for the model component. The **Geometry** node also contains some general settings for the geometry such as the length unit.

Adding a Component Geometry

To add a new Component, which includes a Geometry node, right-click the root node of the **Model Builder**, and select **Add Component** (or use the **Model Wizard** to create a new sequence as described in Creating a New Model). A **Component** node is added to the **Model Builder** containing a **Geometry 1** (if it is the first Component) feature node with the start of a geometry sequence containing only a **Form Union** node (see Figure 7-1 for an example).

See Geometric Primitives, Geometry Operations and Virtual Geometry and Mesh Control Operations for descriptions of the geometry features. The tables in each section link to the individual feature settings window descriptions.

To add features to a geometry sequence, use the buttons in The Geometry Toolbar or right-click The Geometry Node in the **Model Builder** and then select one of the available options.

To open the settings window for a geometry, click the **Geometry** node in the Model Builder and adjust the following settings sections.

UNITS

Select the **Scale values when changing units** check box to scale the values for the geometric dimensions so that the geometric objects keep their physical size. The default setting is to not scale the values when changing units; the program then interprets the values for the geometric dimensions using the new units for length and angle. The values themselves do not change.

From the **Length unit** list select the length unit to use in fields for lengths and for visualization of the geometry. You can override the unit using the unit syntax to specify the length unit (for example, `13[mm]`). When solving the model, all lengths are converted to the base unit for length. If you change the unit, COMSOL Multiphysics converts all pure numeric values in fields for lengths to the new unit, if you have selected the **Scale values when changing units** check box (see above).

For information about available length units and prefixes, see Specifying Model Equation Settings.

Angular Unit

From the **Angular unit** list choose to use radians or degrees as the angular unit to use in fields for angles. You can override the unit by entering, for example, `0.3[rad]`. The program assumes that numeric inputs and outputs of trigonometric functions are in radians. If you change the unit, all pure numeric values in fields for angles are converted to the new unit, if you have selected the **Scale values when changing units** check box (see above).

ADVANCED

Geometry Representation (3D Only)

This list is only visible if you have a license for the CAD Import Module. The **Geometry representation** list controls which kernel (geometric modeler) that COMSOL uses to represent and operate on the geometry objects: the CAD Import Module's kernel (Parasolid) or COMSOL's own kernel.

- If you choose **CAD Import Module kernel** (requires the CAD Import Module), all objects and operations that support the CAD Import Module's kernel use it. For example, Work Plane, Extrude, and Revolve operations do not support this kernel.
- If you choose **COMSOL kernel**, all objects are represented using COMSOL's kernel.

When you change the **Geometry representation** setting, all nodes that support the CAD Import Module's kernel are marked as edited with an asterisk (`*`) in the upper-right corner the node's icon. To rebuild the geometry using the new kernel, click the **Build All** button (▦).

When you create a new model, its default geometry representation is controlled by the preference setting **Geometry>Geometry representation>In new models**.

When you open an existing model, you normally use the geometry representation used in the model. To always get the possibility to convert the geometry to the COMSOL kernel, change the preference setting **Geometry>Geometry representation>When opening an existing model** to **Convert to COMSOL kernel**.

Default Relative Repair Tolerance

This is the default value that is used when you add a feature that has a **Relative repair tolerance** field (for example, Boolean operations and conversions). Changing the **Default relative repair tolerance** does not affect the tolerances in existing features. Adjust the **Relative repair tolerance** if you experience problems with a Boolean operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

Automatic Rebuild

The **Automatic rebuild** check box controls if the geometry sequence is automatically rebuilt when clicking on a node in the model tree outside the geometry sequence. The default value is controlled by the preference setting **Geometry>Automatic rebuild>Default in new geometries**. Select the **Automatic rebuild** check box to always rebuild the geometry, or clear it to prevent any automatic rebuilding of the geometry.

Plane Geometry Node

Under a **Plane Geometry** node ( ) you define and create the geometry sequence for a work plane. The **Plane Geometry** node also contains some settings for the visualization of the work plane's geometry.

Click the **Plane Geometry** node to open the Work Plane Modal Toolbar.

VISUALIZATION

Select the **Draw on work plane in 3D** check box to create the work plane geometry on the work plane in a 3D view (see Drawing on a 2D Work Plane in 3D). The default setting is to display the work plane in the **Graphics** window as a separate 2D geometry.

Under **In-plane visualization of 3D geometry**, specify how to visualize 3D objects in the work plane (as blue curves and points) by selecting one or more of the following check boxes (all of them are selected by default):

- **Coincident entities (blue)**—Show edges and points (in a pure blue color) that lie in the work plane.
- **Intersection (cyan)**—Show the intersection of 3D geometry and the work plane (in cyan).
- **Projection (light blue)**—Show the projection of all edges and points onto the work plane (in light blue).

Creating a Geometry Sequence

There are a variety of ways to add and build geometry nodes—use the buttons on the toolbars, right-click the Geometry node to add items from the context menu, copy and paste geometry features, insert an existing geometry sequence from another MPH file, or import a geometry.

The *Introduction to COMSOL Multiphysics* includes a tutorial to learn how to build the busbar geometry. See Appendix A—Building a Geometry or the printed copy included with COMSOL Multiphysics.

USE THE TOOLBARS AND CONTEXT MENUS

Use the buttons in The Geometry Toolbar. The buttons are available for almost all drawing primitives (in 2D/1D) as well as operations and conversions acting on geometry objects.

You can also select objects or geometric entities in the **Graphics** window and then click a button or choose a context menu item. The selected objects/entities are then an input to the created geometry operation feature. To access all the features from the context menu, right-click The Geometry Node in the Model Builder and then select one of the available options.

However the geometry is added to the sequence, define the node properties in the settings window. In numerical fields you can enter expressions that contain parameters defined in **Parameters** under **Global Definitions** in the Model Tree to parameterize the geometry. Click the **Build Selected** button ( ) in the settings window to see the geometry objects that result.

COPY AND PASTE GEOMETRY OBJECTS

When using the copy/paste functionality, the copy initially contains the same data as the copied node, but there is no future connection between the two nodes. For example, if the original node is changed, it has no effect on the second node that was copied. To keep a link between nodes, use the Transforms Copy feature instead.

With the standard copy and paste method, a copy of the geometry object (a rectangle or sphere, for example) is inserted into the same geometry sequence, or another geometry sequence in the same Component, and is added after the current feature of the selected geometry sequence. The copy feature can also be used for Work Plane geometry sequences.

!

The copied object must be pasted under a Component with the same space dimension. For example, a Sphere can only be pasted into a 3D Component model.

Copying, Pasting, and Duplicating Nodes

INSERT A SEQUENCE

To insert a geometry sequence from an MPH-file:

- On the **Geometry** toolbar, click **Insert Sequence** ( ), or
- Right-click the **Geometry** node and select **Insert Sequence** ( ) from the context menu.

Then browse to a file name and click **Open**. The file is scanned for geometry sequences having the right space dimension. If there is just one such sequence, its nodes are inserted into the geometry sequence after the current node. If the file contains more than one such sequence, a dialog box opens. Select the geometry sequence from the list of available sequences. Finally click **OK**. The nodes in the selected sequence are inserted into the geometry sequence after the current node.

If the geometry sequence contains references to functions or parameters, those functions and parameters are also inserted in the model under **Global Definitions**, or in case of nonglobal functions, under **Definitions** in the same Component the geometry sequence is located.

Functions and parameters are inserted even if a function or parameter with the same name already exists in the model. You have to manually resolve any conflicts before the geometry sequence can be built.

IMPORT A GEOMETRY

To import an existing geometry:

- On the **Geometry** toolbar, click **Import** ( ), or
- Right-click the **Geometry** node and select **Import** ( ) from the context menu.

Then in the **Import** settings window, click **Browse**. Navigate to the geometry file and double-click it. Then click **Import**.

You could also Export an existing geometry from another model and then import it. See Exporting a Geometry for information.

For example, in the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click `virtualgeom_demo_2.mphbin`.

The location of the file varies based on the installation. For example, if the installation is on your hard drive, the file path might be similar to `C:\Program Files\COMSOL44\models`.

- Import
- Exporting a Geometry

Editing and Building Geometry Nodes

THE CURRENT NODE IN GEOMETRY SEQUENCES

Once a geometry node is added, it is inserted in the sequence after the *current node*. To indicate the current node, it displays with a quadratic frame around its icon. When you have added a node, it becomes the current node, but COMSOL Multiphysics does not build it automatically. If you select a node and build it, this node becomes current. The frame is green ( ) to show that the current node is built. If the current node needs to be rebuilt the frame is yellow ( ). See Dynamic Nodes in the Model Builder for examples that show these icon additions for visual feedback about a node's status.

Adding a Node at an Arbitrary Position

To add a node after an existing node, first select the existing node and then click **Build Selected** ( ), or right-click the existing node and select **Build Selected**. The selected node then becomes current. Then add the new node.

To add a node before an existing node, first select the existing node, right-click the existing node and select **Build Preceding** ( ). Then add the node.

EDITING A NODE

To edit a node, select it in the model tree and make changes in the settings window. Nodes that you have edited display with an asterisk (*) at the upper-right corner of their icons in the **Model Builder** window. Nodes that depend on the edited node display with a yellow triangle at the lower-right corner of the node's icon to indicate that they need to be rebuilt. To see the result of your edits in the **Graphics** window, you need to build the node. You can do this in two ways:

- Click the **Build Selected** button ( ) in the settings window, or right-click the node in the tree and select **Build Selected**. This builds all nodes (if needed) from the first up to the selected node.

- Click the **Build All Objects** button (▦) in the settings window. This builds all nodes in the geometry sequence above the **Form Union/Assembly** node (if needed).
- Click the **Build All** button (▦) in the settings window or the toolbar, or right-click the **Geometry** node in the tree and select **Build All**. This builds all nodes in the geometry sequence (if needed).

AUTOMATIC BUILDING OF GEOMETRY NODES

In some situations, COMSOL Multiphysics builds geometry nodes directly, such as when you:

- Add a node, the software builds the current node (and all its preceding nodes) before showing the settings window for the new node.
- Generate a mesh or solve the model, the software builds the finalized geometry; that is, it builds all nodes if needed.
- Select a node that uses the finalized geometry, like physics and mesh nodes, the software builds the finalized geometry if **Automatic rebuild** is selected in the geometry node settings.
- Open a CAD Defeaturing tools settings, the software builds the current node (and all its preceding nodes).

DELETE NODES

- To delete selected nodes, right-click the nodes and select **Delete** (✖) or press Del (the Delete key). Confirm the deletion of nodes for it to take effect.
- To delete a geometry right-click the **Geometry** node in the **Model Builder** and select **Delete Sequence** ( ). You cannot use the **Undo** command.
- To delete geometry objects or entities, in the **Model Builder**, right-click **Geometry** and select **Delete Entities** (🗑). Or select objects in the **Graphics** window, and click the **Delete** button (✖) in the Graphics toolbar.

  If you use the **Delete** button to delete objects, COMSOL Multiphysics deletes the selected objects that correspond to primitive features by deleting their nodes from the geometry sequence. If you delete objects that do not correspond to primitive features or if you delete geometric entities a **Delete Entities** node appears in the sequence.

Undo is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

Exporting a Geometry

You can export geometry objects to a COMSOL binary file (`.mphbin`) or text file (`.mphtxt`).

2D geometry objects can also be exported to a DXF file.

3D geometry objects can also be exported to an STL file. If you have a license for the CAD Import Module, 3D geometry objects can also be exported to Parasolid files (`.x_t`, or `.x_b`) or ACIS files (`.sat`, or `.sab`).

To export an existing geometry to file:

- On the **Geometry** toolbar click **Export** ( ), or
- Right-click the **Geometry** node and select **Export** ( ) from the context menu.

Then select a file type among the available formats in the **File type** list and enter a file name including the path in the **Filename** field (or click **Browse** to specify the filename).

For STL file export you can select objects, domains, or boundaries to export. For the other file types you select the objects to export by first clicking the **Export selected objects** button and then adding the objects to export to the **Selected object** list, or click the **Export result of Form Union/Assembly** button to export the result of the Form Union/Assembly operation.

For a geometry subsequence and for a Plane Geometry of a work plane, the **Export result of Form Union/Assembly** option does not exist, so that button and the **Export selected objects** button are not available. Instead, just add objects to export to the **Selected object** list.

Click **Export** to export the selected geometry to the specified file. A confirmation message appears in The Messages Window.

The `.mphbin` and `.mphtxt` formats do not contain unit information. When the exported file is imported into a geometry with a different length unit, you can use a Scale feature to scale the imported objects to the correct size.

To export a geometry to use in an earlier version of COMSOL in the COMSOL format, select a version from the **Compatible with version** list.

- Also see Import for details of how to import CAD file formats.
- Creating a Geometry Sequence

Measuring Geometry Objects

To measure a set of geometric objects or entities selected in the **Graphics** window, click the **Measure** button (📏) on the **Geometry** toolbar (this button is also available on the **Mesh** toolbar). The result appears in The Messages Window:

- If objects are selected, the number of entities and objects is displayed.
- If domains are selected, the total volume/area/length and boundary area/length is displayed.
- If faces/edges are selected, the total area/length is displayed.
- If a single vertex is selected, the coordinates are displayed.
- If two vertices are selected, the distance is displayed.

Another way to measure geometry objects, is to right-click the **Geometry** node and select **Measure** (📏) from the context menu. This opens the **Measure** window, which has the following contents:

The **Measure** window is a tool to measure geometry objects and entities. You can, for example, measure the volume, area, or length of a selected domain, face, or edge. Also view the coordinates of a vertex, the distance between two vertices, or the number of entities and the geometry representation (requires a license for the CAD Import Module) of an object.

GEOMETRY TYPE

From the **Type of geometry** list, select to measure geometry objects (the default) or the finalized geometry.

SELECTION

From the **Geometric entity level** list, select **Object**, **Domain**, **Boundary**, **Edge**, or **Point**. Then select some objects or entities of the selected type to add to the list.

MEASUREMENTS

Under **Measurements** you find information about these objects or entities. For objects, this section contains the total number of domains, boundaries, edges, and points, and the geometry representation. For domains, boundaries, and edges, their total volume, area, or length appears. If you select a point, its coordinates are shown. If you select two points, the distance is shown.

You can add a **Mass Properties** node to define variables for mass properties such as the total volume, total mass, and the center of mass.

The Form Union/Assembly Node—Uniting the Geometry

To unite the geometry, COMSOL Multiphysics evaluates the geometry sequence from the top down. The final node in the geometry sequence (before any virtual operations, if present), the **Form Union/Assembly** node, determines how to form the geometry that is used for meshing and analysis (possibly after also applying virtual operations). There are two methods to form the geometry, which also determine the name of the node: **Form Union** or **Form Assembly** (). There are some differences and aspects to consider when choosing a method:

- The default method is to form a *union*. The software then forms a union from all geometry objects that the geometry sequence contains or creates. The union is divided into domains separated by boundaries according to the participating geometry objects. You can mesh the entire geometry and model the physics by assigning material properties, boundary conditions, and other data for the model. It is also possible but often not necessary to specify boundary conditions on interior boundaries between domains in the geometry. By default, COMSOL ensures continuity in the physics fields across interior boundaries. See also Removing Interior Boundaries below).

When forming a union for axisymmetric models, COMSOL removes all parts of the geometry from the r<0 half plane.

- The alternative method is to form an *assembly*. The software then treats the geometry as a collection of parts, where each geometry object represents a part. This means that you must use pairs to connect boundaries where a field is continuous, but it also makes is possible to use special pair conditions for applications such as contact modeling. By default, pairs are created automatically when forming an assembly. An assembly can also be useful for meshing each part independently in, for example, thin geometries with high aspect ratios. Another case where you need to use an assembly is when the geometry is too complex for forming a union, which might be the case when importing an assembly geometry from CAD data.

The **Form Union/Assembly** node () ends each geometry sequence in 1D. In 2D and 3D, it is possible to add virtual operation nodes and selection nodes after that node. In the Model Tree, its label is **Form Union** or **Form Assembly** depending on its settings. By default, it unites all geometry objects into a single geometry object (this is the **Form Union** variant). You cannot delete or disable the **Form Union/Assembly** node. When you leave the geometry sequence to define materials or physics, the **Messages** window provides information about the method (forming a union or an assembly) and about the number of geometric entities (domain, boundaries, and so on) in the geometry.

FORM UNION/ASSEMBLY

The default method, **Form a union**, forms a union of all geometry objects. Select **Form an assembly** from the **Action** list if you do not want the geometry objects to be united. The program then forms the geometry by collecting the objects as parts in an *assembly object*. If you form an assembly, select the **Create imprints** check box to get imprints of the parts of the assembly that touch each other. An imprint of a usually smaller part's boundary on an adjacent larger part's boundary inserts points on the boundary in 2D and creates edges on the boundary in 3D. Creating imprints can be useful when you need identical matching meshes on both parts' boundaries or when you want to split the larger boundary so that it contains a segment or area that matches the smaller boundary. Select the **Create pairs** check box (selected by default) to generate pairs corresponding to the parts of the assembly that touch each other. Select the **Split disconnected pairs** check box to generate one pair for each connected set of boundaries. Clear the **Split disconnected pairs** check box to generate one pair for each pair of parts that touch each other. From the **Pair type** list, select **Identity pair** (the default) to generate identity pairs, which makes it possible to connect the physics fields across the assembly parts' boundaries, or **Contact pair** to generate contact pairs. The contact pairs are only useful for contact modeling in structural mechanics and require a license for the Structural Mechanics Module or the MEMS Module.

Adjust the value in the **Relative repair tolerance** field (default value: $1 \cdot 10^{-6}$) if you experience problems with the Form Union/Assembly operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of all geometry objects. Geometric entities with a distance less than the absolute repair tolerance are merged.

In 2D and 3D, virtual operation nodes and selection nodes can appear after the **Form Union/Assembly** node.

Using Geometry Subsequences

If you consider the geometry sequence as a computer program, a *geometry subsequence* corresponds to a subroutine, function, method, or procedure in a programming language. In other words, a geometry subsequence is a geometry sequence whose input is a set of arguments (having numerical values) and whose output is a set of geometry objects.

You can call the subsequence (that is, create an instance of the subsequence) with new values of the arguments. You can call it several times, and the calls can be nested.

CREATING A GEOMETRY SUBSEQUENCE

The geometry subsequences appear under Global Definitions in the Model Tree, so they are not attached to a specific model component. To create a subsequence, right-click **Global Definitions**, and then select **3D**, **2D**, or **1D** from the **Geometry Subsequences** submenu. Beneath the added geometry subsequence node there are two special nodes:

- **Arguments**. Enter the names of the arguments in the **Name** column. In the **Expression** column, enter their default values as expressions in terms of numerical values. The corresponding values appear in the **Value** column. Optionally, enter descriptions for the arguments in the **Description** column.
- **Parameters**. Enter the names of the local parameters in the **Name** column. In the **Expression** column, enter the corresponding definitions; those definitions can depend on arguments and other local parameters. The corresponding values appear in the **Value** column. Optionally, enter descriptions for the local parameters in the **Description** column.

Within the subsequence you can use arguments and local parameters in expressions, just like you use global parameters in an ordinary geometry sequence. There is no support for using global parameters within a subsequence. You add features to the subsequence as usual. There is also a **View** node with view settings below the geometry feature nodes.

CALLING A GEOMETRY SUBSEQUENCE

In a geometry sequence or subsequence you can add a call to a geometry subsequence by right-clicking the **Geometry** node and selecting **Programming>Subsequence Call**. This adds a **Subsequence Call** node (), whose purpose is to build an instance of the subsequence with new values of its arguments. You can also change the position and orientation of the resulting geometry objects. See Subsequence Call for details.

A **Subsequence Call** node can optionally call a local subsequence instead of a subsequence under **Global Definitions**. In this case, the local subsequence appears beneath the **Subsequence Call** node.

There is no associativity for the output objects of a geometry subsequence. That is, if you make a selection that consists of geometric entities from these output objects, the selection becomes empty when you rebuild the geometry after a change in the subsequence or in the arguments. To overcome this problem, define the selections within the subsequence. These selections also become available in the calling sequence. You can also unite selections from several Subsequence Call nodes using a cumulative selection; see Subsequence Call.

DEBUGGING A GEOMETRY SUBSEQUENCE

Sometimes you want to inspect what happens when you build a Subsequence Call. To do so, right-click the **Subsequence Call** node, and select **Step Into**. This builds all preceding features and shows a copy of the called subsequence beneath the **Subsequence Call** node. The copy is identical to the subsequence, except that the arguments have different values. You can now build features in the copy. You can also try out modifications in the copy. You can then apply the modifications in the subsequence.

When you build the Subsequence Call, you lose the changes you have made in the copy. If you want to keep your changes, switch to using a local sequence by selecting **Local subsequence** in the **Subsequence** list.

If an error occurs when you build the Subsequence Call, a copy of the called subsequence appears automatically so that you can locate the error.

Geometric Primitives

The geometric primitives provide building blocks of basic geometric shapes for creating geometries in 1D, 2D, and 3D. The features in Table 7-1 are also available as buttons on the The Geometry Toolbar, sometimes from the **More Primitives** (3D, 2D, or 1D) menu. You can combine and operate on all geometric primitives using Boolean operations and other Geometry Operations.

TABLE 7-1: 1D, 2D, AND 3D GEOMETRY PRIMITIVES AND GEOMETRY TOOLBAR BUTTONS

ICON	NAME	SPACE DIMENSION	ICON	NAME	SPACE DIMENSION
	Bézier Polygon	2D, 3D		Interval	1D
	Block	3D		Parametric Curve	2D, 3D
	Circle	2D		Parametric Surface	3D
	Cone	3D		Point	1D, 2D, 3D
	Cylinder	3D		Polygon	2D, 3D
	Eccentric Cone	3D		Pyramid	3D
	Ellipse	2D		Rectangle	2D
	Ellipsoid	3D		Sphere	3D
	Helix	3D		Square	2D
	Hexahedron	3D		Tetrahedron	3D
	Interpolation Curve	2D, 3D		Torus	3D

If you want to refer to the domains, boundaries, edges, or points (geometric entities) in the Component, COMSOL can create selections for all geometric entities that a geometric primitive consists of. It is also possible to create selections in the geometry nodes for the resulting geometry objects from such operations. See Creating Selections From Geometric Primitives and Operations. For some geometric primitives (blocks and spheres, for example) you can add layers for creating sandwich structures or, for example, layers of concentric spheres.

Bézier Polygon

A **Bézier Polygon** ( ) consists of a sequence of connected line segments, quadratic Bézier curves (for example circular arcs), and cubic Bézier curves. See About Rational Bézier Curves below for some information about Bézier curves in general.

To create a Bézier polygon, you can either:

- Right-click a 2D **Geometry** node and select **Bézier Polygon** ( ). For a 3D model, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Bézier Polygon** ( ). Then enter the properties of the Bézier polygon in the settings window, or
- On a 2D **Geometry** toolbar, click the **Draw Line** button ( ), **Draw Quadratic** ( ), and **Draw Cubic** ( ) buttons to draw in the Graphics window. Also see Drawing Geometric Primitives.

> To draw a polygon consisting of line segments or Bézier curves, first click one of the buttons. Then click the control points of the segments in the Graphics window. Click one point for each linear segment, two points for each quadratic segment, and three points for each cubic segment. If you want to switch segment type, click one of the buttons and then click some more control points. Close the polygon by right-clicking anywhere in the Graphics window. Then, a solid Bézier polygon appears, and a corresponding Bézier Polygon node appears in the geometry sequence. If you want to modify the polygon (for instance, change from solid to curve) you can edit the Bézier Polygon node by clicking it to display its settings window.

When you have added a node or finished drawing the Bézier polygon in the Graphics window, you can use the following sections to define it or fine tune it.

GENERAL

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the Bézier polygon is a solid object (only available in 2D) or a closed or open curve object. If you choose **Solid** or **Closed curve**, the software automatically adds a line segment if needed to close the polygon.

> When using the Geometry toolbar, the **Draw Solid** button ( ) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting a **Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

POLYGON SEGMENTS

Define the Bézier polygon by adding curve segments to the list of segments. Choose linear segments, quadratic segments, or cubic segments. Delete segments by selecting them and clicking **Delete**. To edit a segment, select it in the list. When editing the last segment, click **Close Curve** to make the last control point coincide with the first control point of the first segment.

Linear Segments

To add a linear segment, click **Add Linear**. Specify the start of the linear segment on the first row of coordinates under **Control points**. Specify the end of the linear segment on the second row of coordinates.

Quadratic Segments

To add a quadratic segment, click **Add Quadratic**. Specify the coordinates of the three control points on rows under **Control points**. Add the weights of the control points under **Weights**. The default weights—$1,\ 1/(\sqrt{2})$, and 1—correspond to a circular arc if the control points are three corners of a square.

Cubic Segments

To add a cubic segment, click **Add Cubic**. Specify the coordinates of the four control points on each row under **Control points**. Add the weights of the four control points under **Weights**. Cubic segments with self-intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the Bézier polygon consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

ABOUT RATIONAL BÉZIER CURVES

A rational Bézier curve is a parameterized curve of the form

$$\mathbf{b}(t) = \frac{\sum_{i=0}^{p} \mathbf{b}_i w_i B_i^p(t)}{\sum_{i=0}^{p} w_i B_i^p(t)}, \quad 0 \le t \le 1$$

where the functions

$$B_i^p(t) = \binom{p}{i} t^i (1-t)^{p-i}$$

are the *Bernstein basis* functions of *degree* p; $\mathbf{b}_i = (x_1, \dots, x_n)$ are the control points of the n-dimensional space; and w_i are the weights, which should always be nonnegative numbers. The end-point interpolation property corresponds to $\mathbf{b}(0) = \mathbf{b}_0$ and $\mathbf{b}(1) = \mathbf{b}_p$. Another useful property of the rational Bézier curves is that the direction of the tangent vector at $t = 0$ and $t = 1$ is determined by the vectors $\mathbf{b}_1 - \mathbf{b}_0$ and $\mathbf{b}_p - \mathbf{b}_{p-1}$, respectively. That is, the curve is always tangent to the line connecting the control points $\mathbf{b}_0$ and $\mathbf{b}_1$ and the line connecting $\mathbf{b}_{p-1}$ and $\mathbf{b}_p$. When joining curves at end points, aligning the (nonzero) tangent vectors assures tangential continuity. This technique produces visually smooth transitions between adjacent curves.

Quadratic Curves (Conic Sections)

Rational Bézier curves of degree 2 can represent all conic sections: circles, ellipses, parabolas, and hyperbolas. Elliptical or circular curve segments are often called arcs. The conic sections are also called quadric curves or *quadrics*. Because the parameter t is constrained to be in the interval [0, 1], only a segment of the conic section is represented. A 2nd degree curve consists of three control points and three weights. There is a simple rule for classifying a 2nd degree curve if the end point weights are set to 1, only allowing the central weight w_1 to vary: if $w_0 = w_2 = 1$, then $0 < w_1 < 1$ gives ellipses, $w_1 = 1$ gives parabolas, and $w_1 > 1$ gives hyperbolas. For a fixed control polygon, at most one value of w_1 (among the ellipses generated by letting $0 < w_1 < 1$) gives a circle segment. For example, a quarter of a full circle is generated by a control polygon with a right angle and with a central weight of $1/\sqrt{2}$.

Cubic Curves

Rational Bézier curves of degree 3 (cubic curves) have more dynamic properties than conic section curves. A cubic curve has four control points and four weights, making it possible to create a self-intersecting control polygon or a zigzag control polygon. A self-intersecting polygon can give rise to a self-intersecting curve (loop). Self-intersecting curves and cusps are not supported.

A zigzag control polygon generates an S-shaped curve containing a point of inflection where the tangent line lies on both sides of the curve.

Block

To create a block (box), on the 3D **Geometry** toolbar, click **Block** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the block using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the block is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the edge lengths in the **Width**, **Depth**, and **Height** fields. With the default axis (representing the z-axis) and no rotation, the width, depth, and height correspond to the dimensions in the x-, y-, and z-directions, respectively.

POSITION

Enter the position of the block using the **x**, **y**, and **z** fields. From the **Base** list, choose **Center** if the block is centered about the position, or choose **Corner** if the block has one corner in this position.

AXIS

Specify the direction of the block's third axis—that is, the direction of the edges corresponding to the height. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector in the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the block's third axis in the **Rotation** field. When this angle is zero (the default), the block's second axis is parallel to the xy-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. Specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes under **Layer position** to specify where to apply the layers (see the graphics to the right of the check boxes to see the definitions of the left, right, front, back, bottom, and top sides of the block).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the block consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Circle

To create a circle or disk, you can either:

- Right-click the 2D **Geometry** node and add a **Circle** node to the sequence then define it in the settings window, or
- On the 2D **Geometry** toolbar, from the **Circle** (⊙) menu, select **Circle (Corner)** (○). Then draw the circle in the Graphics window. Also see Drawing Geometric Primitives.

> To draw a circle for a 2D model, click Circle or Circle (Corner). Then, click the circle’s center (or one corner of the circle’s bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the circle’s bounding box. When you release the mouse button, a solid circle appears, and a Circle node appears in the geometry sequence.

When you have added a node or finished drawing the circle in the Graphics window, you can use the following section to define it or fine tune it.

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the circle is a solid object (disk) or a curve object.

> When using the geometry toolbar, the **Draw Solid** button (⌐) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

SIZE AND SHAPE

Define the circle’s radius in the **Radius** field. Enter a sector angle (in degree) for a circle sector in the **Sector angle** field. The default value is 360 degrees for a full circle.

POSITION

Enter the position of the circle using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the circle is centered about the position, or choose **Corner** if a surrounding box has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the position in the **Rotation** field. The default angle is 0 degrees.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric circles. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the circle consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Cone

To create a right circular cone or cone frustum (conical frustum, truncated cone), on the 3D **Geometry** toolbar, click **Cone** (▷). By adding a **Cone** feature you can create the part of a cone contained between two circular bases without going through an apex. You can also right-click the **Geometry** node to add this node from the context menu. Enter the properties of the cone using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the cone is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the cone in the **Bottom radius**, **Height**, **Specify top size using**, and **Semi-angle** or **Top radius** fields. From the **Specify top size using** list select **Radius** to specify the cone top size using the **Top radius** field. The top radius must be a positive value or 0 for a cone with a sharp apex. From the **Specify top size using** list select **Angle** (the default setting) to specify the cone top size using the **Semi-angle** field. The semi-angle is the angle a cone makes with the vertical axis. For the default cone with a bottom radius and height of 1, the default semi-angle is roughly 26.565 degrees, which makes the radius of the second (top) basis 1/2. For the default radius and height the maximum semi-angle is 45 degrees (for a cone with a sharp apex). The maximum semi-angle depends on the values for the radius and height. The semi-angle must be larger than –90 degrees. Setting the semi-angle to 0 makes the cone into a cylinder.

POSITION

Enter the position of the cone using the **x**, **y**, and **z** fields. This is the center of the bottom circle.

AXIS

Specify the direction of the cone’s axis. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cone’s local coordinate system is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers to one or more sides of the cone. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cone consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Cylinder

To create a right circular cylinder, on the **Geometry** toolbar, click **Cylinder** (▭). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the cylinder using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the cylinder is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the cylinder in the **Radius** and **Height** fields.

POSITION

Enter the position of the cylinder using the **x**, **y**, and **z** fields. This is the center of the bottom circle.

AXIS

Specify the direction of the cylinder's axis. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cylinder's local coordinate system is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cylinder consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Eccentric Cone

To create an eccentric (oblique) cone or cone frustum with elliptic base, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Eccentric Cone** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the eccentric cone using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the eccentric cone is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the eccentric cone in the **a-semiaxis**, **b-semiaxis**, **Height**, **Ratio**, **Top displacement 1**, and **Top displacement 2** fields. The bottom of the cone is an ellipse with semiaxes given in the **a-semiaxis** and **b-semiaxis** fields. The **Height** field determines the height of the cone frustum. The **Ratio** field controls the ratio between the perimeters of the top and bottom ellipses. To get an oblique cone, use the **Top displacement** fields to specify the displacement of the top ellipse's center relative to the bottom ellipse's center, in the cone's local coordinate system.

POSITION

Enter the position of the eccentric cone using the **x**, **y**, and **z** fields. This is the center of the bottom ellipse.

AXIS

Specify the direction of the third axis of the cone’s local coordinate system—that is, the normal to the base ellipse. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cone’s local coordinate system is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the eccentric cone consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Ellipse

To create a ellipse, you can either:

- Right-click the **Geometry** node and add an **Ellipse** node to the sequence then define it in the settings window, or
- On the 2D **Geometry** toolbar, from the **Circle** (⊙) menu, select **Ellipse** (⬭) or **Ellipse (Corner)**(⬭).Then draw the ellipse in the Graphics window. Also see Drawing Geometric Primitives.

To draw an ellipse for a 2D model, click Ellipse or Ellipse (Corner). Then, click the ellipse’s center (or one corner of the ellipse’s bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the ellipse’s bounding box. When you release the mouse button, a solid ellipse appears, and an Ellipse node appears in the geometry sequence.

When you have added a node or finished drawing the ellipse in the Graphics window, you can use the following section to define it or fine tune it.

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the ellipse is a solid object or a curve object.

When using the geometry toolbar, the **Draw Solid** button (⌐) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

SIZE AND SHAPE

Define the ellipse’s semiaxes in the **a-semiaxis** and **b-semiaxes** fields. Enter a sector angle (in degree) for an ellipse sector in the **Sector angle** field. The default value is 360 degrees for a full ellipse.

POSITION

Enter the position of the ellipse using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the ellipse is centered about the position, or choose **Corner** if a surrounding box has one corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the base point in the **Rotation** field. The default angle is 0 degrees.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric ellipses. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, edges, and points) that the ellipse consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the Model Tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Ellipsoid

To create an ellipsoid, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Ellipsoid** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the ellipsoid using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the ellipsoid is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the semiaxes of the ellipsoid in the **a-semiaxis**, **b-semiaxis**, and **c-semiaxis** fields.

POSITION

Enter the position of the ellipsoid's center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the ellipsoid's third axis—that is, the principal axis corresponding to **c-semiaxis**. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector in the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the ellipsoid's third axis in the **Rotation** field. When this angle is zero (the default), the ellipsoid's second axis is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric ellipsoids. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the ellipsoid consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Helix

To create a helix (coil) with a circular cross section, on the **Geometry** toolbar click **Helix** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the helix using the sections in the settings window.

To create a helix with a noncircular cross section, define the cross section using a work plane. Define the helix centerpoint as a 3D curve using a **Helix** node with a minor radius = 0 or a **Parametric Curve** node, and then use a **Sweep** node to sweep the cross section from the work plane along the curve to create the helix.

OBJECT TYPE

From the **Type** list, select **Solid** (the default) to create a solid helix, or select **Surface** to create a hollow helix that consists of surfaces only.

SIZE AND SHAPE

This section contains a number of properties that determine the size and shape of the helix.

The **Number of turns** field contains a positive number. The default value is 3 turns.

There are two radii:

- The **Major radius** (SI unit: m) field is the radius from the center of the helix (the default is 1 m).
- The **Minor radius** field (SI unit: m) is the radius of the cross section (the default is 0.1 m). The **Minor radius** can be zero, in which case a curve object is created. You can use this together with the **Sweep** feature to create helices with noncircular cross sections.

There are two pitches:

- The **Axial pitch** field (SI unit: m) determines the axial distance between similar positions on two consecutive turns of the helix (the default is 0.3 m).
- The **Radial pitch** field (SI unit: m) determines the radial distance between similar positions on two consecutive turns of the helix (the default is 0, which means that each turn has the same radius).

Select **Right handed** or **Left handed** from the **Chirality** list. The *chirality* or handedness of the helix can be either right handed (the default) or left handed. For a right handed helix, a clockwise screwing motion moves the helix away from the observer; for a left handed helix, a clockwise screwing motion moves it toward the observer.

From the **End caps** list, select an option to create the end caps of the helix:

- Select **Parallel to axis** (the default) to create end caps that are parallel to the helix axis.
- Select **Perpendicular to axis** to create end caps that are perpendicular to the helix axis.
- Select **Parallel to spine** to create end caps that are parallel to the spine of the helix.

POSITION

This is the center position for the starting turn of the helix. Enter the coordinates in the **x**, **y**, and **z** fields. The default position is the origin.

AXIS

Select the **Axis type**—**x-axis**, **y-axis**, **z-axis**, **Cartesian**, or **Spherical**.

- Select **x-axis**, **y-axis**, or **z-axis** (the default) to define the axis direction parallel to one of the coordinate axes.
- Select **Cartesian** to define the axis direction using Cartesian coordinates in the **x**, **y**, and **z** fields. The default axis is in the z-direction (0, 0, 1).
- Select **Spherical** to define the axis direction using spherical coordinates θ and φ (angles of inclination and azimuth, respectively) in the **theta** and **phi** fields. The default angles are 0.

ROTATION ANGLE

Rotate the helix around its axis by entering an angle in the **Rotation** field. The default value is 0 degrees.

ADVANCED SETTINGS

By default, the **Twist compensation** check box is selected, which prevents the twisting that would otherwise occur due to nonzero torsion for curves that do not belong to a fixed plane. Twist compensation rotates the base circle during the sweep along the helix curve by an amount equal to the integral of the curve torsion.

Twist compensation affects the position of the vertices on the top side of the helix. Clear the **Twist compensation** check box to turn it off. See Figure 2-6 below.

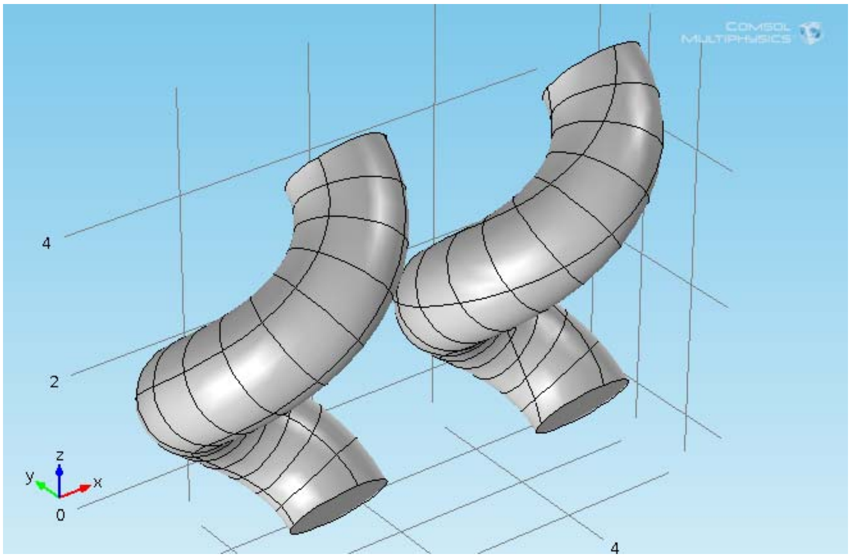


Figure 7-2: The helix on the left has twist compensation (the default). For the helix on the right, twist compensation has been turned off.

From the **Geometry representation** list, select **Spline** (the default) to represent the helix using splines, or **Bézier**, to represent the helix using Bézier curves. The difference is that using Bézier curves, the intersections between the surfaces that form the helix are visible edges, whereas they are hidden when using splines.

The values in the **Relative tolerance** field is a relative tolerance that controls the accuracy of the geometric representation of the helix. The geometric representation is an approximation, which is necessary because it is not possible to exactly represent a helix using NURBS (nonuniform rational basis splines). The default value is 10^{-4} (or 0.01%).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the helix consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Hexahedron

To create a hexahedron bounded by bilinear faces, on the **Geometry** toolbar, click **More Primitives>Hexahedron** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the hexahedron using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the hexahedron is a solid object or a (hollow) surface object.

VERTICES

Define the position, size, and shape of the hexahedron by specifying the coordinates of its vertices. Vertices 1–4 are the vertices of the bottom face in clockwise order. Vertices 5–8 are the vertices of the top face in clockwise order.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the hexahedron consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Interpolation Curve

An interpolation curve consists of a curve that interpolates or approximates a sequence of points. To create an interpolation curve, on the **Geometry** toolbar, from the **More Primitives** (3D or 2D) menu, select **Interpolation Curve** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the interpolation curve using the following sections:

INTERPOLATION CURVE

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the interpolation curve is a solid object (only available in 2D) or a closed or open curve object. If **Solid** or **Closed curve** is selected, a point is automatically added if needed to close the curve, and the curve has continuous first and second derivatives everywhere.

From the **Data source** list, select **Table** to specify the points to interpolate in a table in the settings window. This is the default data source.

From the **Data source** list, select **Vectors** to specify the points to interpolate as vectors (lists) in the fields **x**, **y**, and (3D only) **z**; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. Each field can contain a list of numbers or expressions containing parameters, separated with commas or spaces. Click the **Range** button ( ) to use the **Range** dialog box for specifying the vector of values for each coordinate.

From the **Data source** list, select **File** to read the points to interpolate from a text file. Specify the file name in the **Filename** field or click the **Browse** button. If **Data format** is **Spreadsheet**, the file must be a text file with the number of columns equal to the dimension of the geometry sequence, and one row for each data point. The columns can be separated by space, tab, comma, or semicolon characters. If **Data format** is **Sectionwise**, the file must be in the sectionwise COMSOL postprocessing data format (see Sectionwise Data Format). Click the **Import to Table** button to copy the file contents into the data point table and change the **Data source** to **Table**.

If **Data source** is **File**, changes in the file do not automatically cause the interpolation curve feature to be rebuilt. To rebuild the feature after a change in the file, click the **Rebuild with Current File** button.

In the **Relative tolerance** field, enter the maximum allowed distance between the generated curve and the sequence of points. The default value 0 implies that the curve interpolates all points. If the relative tolerance is larger than 0, the curve does not necessarily interpolate all points, but the first and last points are interpolated.

Curves with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also if two different parts of the curve touch, even if they do not intersect.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of the following types: domains, boundaries, edges, and points) that the interpolation curve consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Interval

To create one or several intervals, you can either:

- Right-click the **Geometry** node and add an **Interval** node to the sequence then define it in the settings window, or
- On the 1D **Geometry** toolbar, click **Interval** ( ). Then draw the interval in the Graphics window. Also see Drawing Geometric Primitives.

When you have added a node or finished drawing the interval in the Graphics window, you can use the following section to define it or fine tune it.

INTERVAL

Enter the coordinates of the endpoints of the interval in the **Left endpoint** and **Right endpoint** fields. To get an object consisting of a sequence of connected intervals, change **Number of intervals** from **One** to **Many**, and enter a comma-separated list of coordinates in the **Points** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all domains and points that the interval consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Parametric Curve

A parametric curve is a curve in 2D and 3D where you use a parameter to define the coordinates of the curve. For example, the coordinates ($s \cdot \cos(s)$, $s \cdot \sin(s)$) for a parameter s that runs from 0 to 10π defines a spiral in 2D. To create a parametric curve, on the **Geometry** toolbar, from the **More Primitives** (3D or 2D) menu, select

Parametric Curve ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the parametric curve using the following sections:

PARAMETER

Define the parameter name in the **Name** field (default name: `s`). Also define the interval for the parameter values in the **Minimum** (default: 0) and **Maximum** (default: 1) fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each spatial coordinate in the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (3D only) **z** fields. To create the spiral described earlier with the parameter *s*, type `s*cos(s)` in the **x** field and `s*sin(s)` in the **y** field.

> Self-intersecting curves are not supported, except for closed curves (that is, when the start and end points coincide).

By default, the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (in 3D) **z** expressions define the coordinates of points on the curve in the standard coordinate system. It is, however, possible to change this using the settings in the **Position**, **Axis** (3D only), and **Rotation Angle** sections. This is useful if you have created a parametric curve with the right shape but want to move it to another position or orientation. These settings can be thought of as defining a local coordinate system in which the parametric curve is defined.

Curves with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also if two different parts of the curve touch, even if they do not intersect. Simple closed curves are allowed though.

POSITION

Enter the position of the local coordinate system origin using the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (3D only) **z** fields.

AXIS

In 3D, enter the axis that you want to rotate the local coordinate system about. The axis can be chosen parallel to one of the coordinate axes or entered in Cartesian or spherical coordinates. The *z*-axis of the local coordinate system is parallel to this axis.

ROTATION ANGLE

Enter the angle you want the local coordinate system to be rotated (default: 0 degrees). In 2D the local coordinate system is rotated about its origin. In 3D, the local coordinate system is rotated about its *z*-axis, which is parallel to the axis defined in the previous section.

ADVANCED SETTINGS

Internally, the software represents the parametric curve by a B-spline, which is computed to approximate the mathematical curve defined by the **x**, **y** in 2D, **r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes, and **x**, **y**, and **z** in 3D expressions. The number of knot points in the spline increases automatically until the curve approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field. The tolerance is measured relative to the space diagonal of the bounding box of the parametric curve.

If the coordinate expressions contain user-defined functions, changes in those functions do not automatically cause the parametric curve feature to be rebuilt. To rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries or edges, and points) that the parametric curve consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Parametric Surface

A parametric surface is a surface in 3D where you use two parameters to define the coordinates of the surface. For example, the coordinates $(s_1\cdot\cos(s_2), s_1\cdot\sin(s_2), s_2)$ for a parameter s_1 that runs from 0 to π, and a parameter s_2 that runs from −1 to 1 define a “twisted rectangle.” To create a parametric surface, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Parametric Surface** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the parametric surface using the following sections:

PARAMETERS

Define the parameter names in the **Name** fields (default names: `s1` and `s2`). Also define the intervals for the parameter values in the **Minimum** (default: 0) and **Maximum** (default: 1) fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each spatial coordinate in the **x**, **y**, and **z** fields. To create the twisted rectangle described earlier with the parameters s_1 and s_2, type `s1*cos(s2)` in the **x** field, `s1*sin(s2)` in the **y** field, and `s2` in the **z** field.

By default, the **x**, **y**, and **z** expressions define the coordinates of points on the surface in the standard coordinate system. It is, however, possible to change this using the settings in the **Position**, **Axis**, and **Rotation Angle** sections. This is useful if you have created a parametric surface with the right shape but want to move it to another position or orientation. These settings can be thought of as defining a local coordinate system in which the parametric surface is defined.

Surfaces with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also to surfaces where one edge touches the surface or another edge, and to surfaces with singular points. If necessary, several parametric surfaces can be combined to overcome this limitation. For example, constructing a cylindrical shell by typing `cos(s1)` in the **x** field, `sin(s1)` in the **y** field, and `s2` in the **z** field, where `s1` runs from 0 to 2π, and `s2` runs from 0 to 1, is incorrect because two edges of the parametric surface touch each other. Instead, use two parametric surfaces, with the same coordinate expressions, and where `s1` runs from 0 to π in the first surface and from π to 2π in the second one.

POSITION

Enter the position of the local coordinate system origin using the **x**, **y**, and **z** fields.

AXIS

Enter the axis that you want to rotate the local coordinate system about. The axis can be chosen parallel to one of the coordinate axes or entered in Cartesian or spherical coordinates. The z-axis of the local coordinate system is parallel to this axis.

ROTATION ANGLE

Enter the angle you want the local coordinate system to be rotated (default: 0 degrees). The local coordinate system is rotated about its z-axis, which is parallel to the axis defined in the previous section.

ADVANCED SETTINGS

Internally, the software represents the parametric surface by a B-spline, which is computed to approximate the mathematical surface defined by the **x**, **y**, and **z** expressions. The number of knot points in the spline increases automatically until the surface approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field. The tolerance is measured relative to the space diagonal of the bounding box of the parametric surface.

If the coordinate expressions contain user-defined functions, changes in those functions do not automatically cause the parametric surface feature to be rebuilt. To rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries—that is, the parametric surfaces—, edges, and points) that the parametric surface consists of. These selections are available in all applicable selection lists (as, for example, **Parametric Surface 1 {geom1_ps1_edg}** for a selection that includes the parametric surface's edges) but do not appear as separate selection nodes. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Point

To create points, you can either:

- For all space dimensions, right-click the **Geometry** node and add a **Point** node to the sequence and then define it in the settings window, or
- On the 1D **Geometry** toolbar, click **Point** (▪). On the 2D **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Point** (▪). Then draw the point by clicking in the Graphics window. Also see Drawing Geometric Primitives. On the 3D **Geometry** toolbar, from the **More Primitives** ( ) menu, you can also select **Point** (▪), which then adds a **Point** node to the sequence. You can then define the point's location in the **Point** node's settings window.

When you have added a node or finished drawing the point in the Graphics window, you can use the following section to define it or fine tune it.

POINT

Define the position of the point by entering its coordinates in fields labeled **x**, **y** (2D and 3D), and **z** (3D); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To get several points, enter a list of coordinates in each of these fields. Separate the coordinates with commas or blanks.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create a predefined selection for the point. This selection is available in all selection lists for points but does not appear as a separate selection node in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Polygon

A polygon consists of a sequence of connected line segments. To create a polygon, right-click a 2D **Geometry** node and select **Polygon** ( ). For a 3D model, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Polygon** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the polygon using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the polygon is a solid object (only available in 2D) or a closed or open curve object. If you choose **Solid** or **Closed curve**, the program adds a line segment if needed to close the polygon.

When using The Geometry Toolbar, the **Draw Solid** button ( ) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

COORDINATES

From the **Data source list**, you can choose from three different data sources for the coordinates:

- Select **Vectors** (the default) to specify the coordinates of the vertices as vectors (lists) in the fields **x**, **y**, and **z** (3D only); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. Each field can contain a list of numbers or expressions containing parameters, separated with commas or spaces. Click the **Range** button ( ) to use the **Range** dialog box for specifying the vector of values for each coordinate.
- Select **Table** to specify the coordinates of the vertices in a table directly in the settings window.
- Select **File** to read vertex coordinate data from a text file where each row represents the *x*, *y*, and (in 3D) *z* coordinates for a vertex in the polygon. Specify the filename in the **Filename** field, or click the **Browse** button. The file must be a text file with the number of columns equal to the dimension of the geometry sequence and one row for each data point. The columns can be separated by space, tab, comma, or semicolon characters. Click the **Import to Table** button to copy the file contents into the data point table and change the **Data source** to **Table**. Changes in the file do not automatically cause the polygon to be rebuilt. To rebuild the node after a change in the file, click the **Rebuild with Current File** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of the following types: domains, boundaries, edges, and points) that the polygon consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Pyramid

To create a rectangular pyramid or pyramid frustum, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Pyramid** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the pyramid using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the pyramid is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the pyramid in the **Base length 1**, **Base length 2**, **Height**, **Ratio**, **Top displacement 1**, and **Top displacement 2** fields. The **Base length** fields determine (default: 1) the side lengths of the bottom rectangle. The **Height** field (default: 1) determines the height of the pyramid frustum. The **Ratio** field (default: 0.5) controls the ratio of the perimeters of the top and bottom rectangles. To get an oblique pyramid, use the **Top displacement** fields (default: 0) to specify the displacement of the top rectangle's center relative to the bottom rectangle's center, in the pyramid's local coordinate system.

POSITION

Enter the position of the pyramid using the **x**, **y**, and **z** fields. This is the center of the bottom rectangle.

AXIS

Specify the direction of the third axis of the pyramid's local coordinate system—that is, the normal to the base rectangle. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the pyramid's local coordinate system (corresponding to **Base length 2**) is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the pyramid consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Rectangle

To create a rectangle, you can either:

- Right-click the **Geometry** node and add a **Rectangle** node to the sequence then define it in the settings window, or
- On the **Geometry** toolbar, from the **Rectangle** (▭) menu, select **Rectangle (Center)** (▣). Then draw the rectangle in the Graphics window. Also see Drawing Geometric Primitives.

To draw a rectangle for a 2D model, click Rectangle or Rectangle (Center). Then, click one corner (or the center) of the rectangle in the Graphics window. Drag the mouse to the desired position of a corner. When you release the mouse button, a solid rectangle appears, and a Rectangle node appears in the geometry sequence.

When you have added a node or finished drawing the rectangle in the Graphics window, you can use the following section to define it or fine tune it.

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the rectangle is a solid object or a curve object.

When using the geometry toolbar, the **Draw Solid** button (⊓) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

SIZE

Define the size of the rectangle in the **Width** and **Height** fields.

POSITION

Enter the position of the rectangle using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the rectangle is centered about the position, or choose **Corner** if the rectangle has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle (default: 0 degrees) about the position the **Rotation** field.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the rectangle consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Sphere

To create a sphere or ball, on the **Geometry** toolbar click **Sphere** (⊕). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the sphere using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the sphere is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the radius of the sphere in the **Radius** field.

POSITION

Enter the position of the sphere's center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the third axis of the sphere's local coordinate system. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the sphere's local coordinate system is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric spheres. You specify the thicknesses and, optionally, names of each layer in the **Layers** table. The outermost layer comes first. The layers are positioned inside the sphere's radius.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the sphere consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Square

To create a square, you can either:

- Right-click the **Geometry** node and add a **Square** node to the sequence then define it in the settings window, or
- On the **Geometry** toolbar, from the **Rectangle** ( ) menu, select **Square** ( ) or **Square (Center)**( ). Then draw the square in the Graphics window. Also see Drawing Geometric Primitives.

To draw a square, click Square or Square (Center). Then, click one corner (or the center) of the square in the Graphics window. Drag the mouse to the desired position of a corner. When you release the mouse button, a solid square appears, and a Square node is added to the geometry sequence.

When you have added a node or finished drawing the square in the Graphics window, you can use the following section to define it or fine tune it.

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the square is a solid object or a curve object.

When using the geometry toolbar, the **Draw Solid** button ( ) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**. See Geometry Drawing Toolbar Buttons for other drawing toolbar buttons.

SIZE

Define the size of the square in the **Side length** field.

POSITION

Enter the position of the square using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the square is centered about the position, or choose **Corner** if the square has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the position the **Rotation** field.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the square consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Tetrahedron

To create a tetrahedron, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Tetrahedron** ( ).You can also right-click the **Geometry** node to add this node from the context menu.

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the tetrahedron is a solid object or a (hollow) surface object.

VERTICES

Define the position, size, and shape of the tetrahedron by specifying the coordinates of its vertices. Vertices 1–3 are the vertices of the bottom face in clockwise order. Vertex 4 is the top vertex.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the tetrahedron consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Torus

To create a torus, on the **Geometry** toolbar, from the **More Primitives** ( ) menu, select **Torus** ( ). You can also right-click the **Geometry** node to add a **Torus**.

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the torus is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the torus in the **Major radius**, **Minor radius**, **Revolution angle** and **Interior faces** fields. The **Major radius** (default: 1) field controls the distance from the center of the cross section to the center of the torus. The **Minor radius** (default: 0.5) field controls the radius of the cross section. To get less than a full revolution (360 degrees, the default), use the **Revolution angle** field. Select the **Interior faces** check box to create cross-sectional faces that partition the domain of a **Solid** torus.

POSITION

Enter the position of the torus' center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the third axis of the torus' local coordinate system—that is, the normal to the plane of directrix circle. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero, the second axis of the torus' local coordinate system is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the torus consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Composite Object (Backward Compatibility)

If you open a model created in the 3.5a version of COMSOL one **Composite Object** node ( ) appears for each nonprimitive geometry object in the model. The **Composite Object** node contains the follow sections:

COMPOSITE OBJECT

If you save the model as a `.java` file, COMSOL uses the filename specified in the **Filename** field to determine the path to a geometry file, containing the geometry object, that appears together with the `.java` file. The software uses this geometry file when you run the resulting `.java` file. By default, the filename has the prefix $FILENAME$. If the filename starts with this prefix, COMSOL stores the geometry file in the same directory as the `.java` file. It is also possible to remove this prefix and specify the full path to the geometry file.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the composite object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Geometry Operations

This section describes the available geometry operations (Work Plane Operations, Boolean Operations, Transform Operations, Conversion Operations, Other Geometry Operations, and Programming). The features in Table 7-2, Table 7-3, Table 7-4, Table 7-5, and Table 7-6,are also available as buttons on the The Geometry Toolbar.

See Virtual Geometry and Mesh Control Operations for examples of how to use virtual geometry operations to remove a short edge and to ignore and collapse edges to prepare the geometry for efficient meshing. You can use several of these operations to also control the mesh.

WORK PLANE OPERATIONS

The following table lists operations available with work planes in 3D geometries:

TABLE 7-2: WORK PLANE RELATED OPERATIONS FOR 3D MODELS

ICON	NAME	DESCRIPTION
	Cross Section	Create a 2D cross section from an intersection between a 3D geometry and a work plane.
	Extrude	Extrude planar objects into 3D.
	Revolve	Revolve planar objects into 3D.
	Sweep	To sweep one or several faces along a spine curve.
	Work Plane	Create a work plane for drawing 2D objects that are embedded in 3D.

BOOLEAN OPERATIONS

Use Boolean operations to create a *composite geometry object* by forming unions, set differences, and set intersections—and combinations of those operations—of existing geometry objects.

The following Boolean operations are available in all space dimensions:

TABLE 7-3: BOOLEAN OPERATIONS

ICON	NAME	SPACE DIMENSION	DESCRIPTION
	Compose	1D, 2D, 3D	Compose geometry objects by specifying Boolean operations using a set formula.
	Difference	1D, 2D, 3D	Select the objects to compose in the Graphics window and click the Difference button to take the selected object with the largest volume (area, length) and subtract the others.
	Intersection	1D, 2D, 3D	Select the objects to intersect in the Graphics window and click the Intersection button to create an intersection of the selected objects.
	Partition	1D, 2D, 3D	Partition geometry objects using a work plane or other geometry objects (tool objects).
	Union	1D, 2D, 3D	Select the objects to unite in the Graphics window and click the Union button to create a union of the selected objects.

TRANSFORM OPERATIONS

You can use the transforms to create rectangular and linear arrays of identical geometry objects and to move, rotate, mirror, and scale geometry objects. Mirroring, moving, rotating, and scaling are *affine transformations* applied to geometry objects. All transforms are available in all space dimensions, except Rotate, which is not applicable for 1D geometries.

TABLE 7-4: GEOMETRY TRANSFORM OPERATIONS

ICON	NAME	SPACE DIMENSION	DESCRIPTION
	Array	1D, 2D, 3D	Create an array of geometry objects.
	Copy	1D, 2D, 3D	Copy geometry objects.
	Mirror	1D, 2D, 3D	Mirror geometry objects in a plane (3D), a line (2D), or a point (1D).
	Move	1D, 2D, 3D	Move geometry objects.
	Rotate	2D, 3D	Rotate geometry objects about a centerpoint.
	Scale	1D, 2D, 3D	Scale geometry objects about a centerpoint.

CONVERSION OPERATIONS

The geometry conversion operations make it possible to, for example, convert a 3D solid to a surface (boundary) object for modeling a shell or other thin structure. You can perform the following geometry object conversions:

For 2D models:

- Convert a solid object into a curve or point object.
- Convert a curve object defining at least one closed domain into a solid object.
- Convert a curve object into a point object.

For 3D models:

- Convert a solid object into a surface, curve, or point object.
- Convert a surface object defining at least one closed domain into a solid object.
- Convert a surface object into a curve or point object.
- Convert a curve object into a point object.

TABLE 7-5: GEOMETRY CONVERSION OPERATIONS

ICON	NAME	SPACE DIMENSION	DESCRIPTION
	Convert to Curve	2D, 3D	Select the objects to convert in the Graphics window and click to unite them and convert them to a curve object.
	Convert to Point	1D, 2D, 3D	Select the objects to convert in the Graphics window and click to unite them and convert them to a point object.

TABLE 7-5: GEOMETRY CONVERSION OPERATIONS

ICON	NAME	SPACE DIMENSION	DESCRIPTION
	Convert to Solid	1D, 2D, 3D	Select the objects to convert in the Graphics window and click to unite them and convert them to a solid object.
	Convert to Surface	3D	Select the objects to convert in the Graphics window and click to unite them and convert them to a surface object.
	Convert to COMSOL	3D	Available with the CAD Import Module. Select the objects to convert in the Graphics window, and then click to convert them to a COMSOL kernel representation. See the *CAD Import Module User's Guide*.
	Split	1D, 2D, 3D	Split one or several objects into their entities.

OTHER GEOMETRY OPERATIONS

For 2D geometry objects, you can use geometry operations such as fillets and tangents to construct the geometry. In all space dimensions you can delete geometry objects or geometric entities (domains, boundaries, edges, and points).

The following table provides an overview of available general geometry operations:

TABLE 7-6: OTHER GEOMETRY OPERATIONS

ICON	NAME	SPACE DIMENSION	DESCRIPTION
	Chamfer	2D	Create chamfers at a set of corners of a 2D geometry object.
	Fillet	2D	Create fillets at a set of corners of a 2D geometry object.
	Tangent	2D	Create a tangent from an edge to another edge or point in a 2D geometry.
	Delete Entities	1D, 2D, 3D	To delete geometric entities (domains, boundaries, edges, or points) from the objects they belong to, or to delete entire geometry objects, select the entities or objects and click the Delete button. If you delete objects corresponding to primitive features, these nodes are removed from the sequence. If you delete other objects or geometric entities, a Delete Entities node is added to the sequence. If instead you use the Delete Entities context menu item, a Delete Entities always displays.
	Edit Object	2D	To edit a 2D object using the settings window, in the Model Builder, right-click the Geometry node and select Edit Object from the context menu. To edit a 2D object using the Graphics window, select the object and click the Edit Object button. See Editing 2D Geometry Objects.
	Import	1D, 2D, 3D	Import geometry objects from a file or from another geometry.

PROGRAMMING

You can use If, Else If, Else, and End If nodes to create If statements that enable or disable other features depending on values of logical conditions in terms of parameters. You can also use a Subsequence Call node to create an instance of a geometry subsequence (see Using Geometry Subsequences) with changed values of its arguments.

TABLE 7-7: PROGRAMMING AND LOGICAL OPERATIONS

ICON	NAME	DESCRIPTION
?	If	Begin an If statement. See If, Else If, Else, End If.
?	Else If	Continue an If statement. See If, Else If, Else, End If.
?	Else	Last alternative in an If statement. See If, Else If, Else, End If.
?	End If	End an If statement. See If, Else If, Else, End If.
↳A	Subsequence Calll	Call a geometry subsequence.

Array

To create a rectangular or linear array of identical geometry objects, on the **Geometry** toolbar, click **Array** (▦). You can also right-click the **Geometry** node to add this node from the **Transforms** submenu. Then enter the properties of the array operation using the following sections:

INPUT

Select the geometry objects that you want to duplicate in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Array** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

SIZE

In 2D and 3D, you get a rectangular (in 2D) or three-dimensional (in 3D) array by default. Enter the number of duplicates in each coordinate direction in the **x size**, **y size**, and **z size** fields; **r size** and **z size** in 2D axial symmetry; **xw size** and **yw size** in work planes.

To create a linear array of objects in 2D or 3D, change **Array type** to **Linear**. Enter the number of duplicates in the **Size** field.

In 1D, enter the number of duplicates in the **Size** field.

DISPLACEMENT

Set the displacement in each coordinate direction in the **x**, **y**, and **z** fields (not all fields are available in 1D and 2D).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Chamfer

To chamfer corners in 2D geometry objects, on the **Geometry** toolbar click **Chamfer** ( ) You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the chamfer operation using the following sections:

POINTS

Select the points (vertices) that you want to chamfer in the Graphics window. They then appear in the **Vertices to chamfer** list. If the geometry sequence includes user-defined selections above the **Chamfer** node, choose **Manual** to select points, or choose one of the selection nodes from the list next to **Vertices to chamfer**.

Click the **Active** button to toggle between turning ON and OFF the **Vertices to chamfer** selections.

DISTANCE

In the **Distance from vertex** field, enter the distance from the vertex to the endpoints of the chamfer segment.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Compose

To create a composite geometry object from other geometry objects using Boolean operations given in a set formula, on the **Geometry** toolbar, click **Compose** ( ). You can also right-click the **Geometry** node to add this node from the **Boolean Operations** submenu. Then enter the properties of the compose operation as a set formula using the following section:

COMPOSE

Select the geometry objects that you want to compose in the Graphics window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Compose** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected solid geometry objects for further geometry operations.

In the **Set formula** field, enter a set formula involving the names of the selected geometry objects—for example, `r1+c1*(c2-r2)`—to take the union of `r1` and the object that is the result of the intersection between `c1` and the set difference where `r2` is subtracted from `c2`. Use the binary operations `+`, `*`, and `-` for set union, set intersection, and set difference, respectively. The precedence of the operators `+` and `-` are the same. The operator `*` has higher precedence. You can override the precedence rules using parentheses. When you change the set formula, the **Input objects** selection is automatically updated.

To create a geometry object without interior boundaries, clear the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the compose operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. The value is relative to the overall dimensions of the geometry. For example, if the dimensions are in meters, the default repair tolerance makes the geometry repair heal gaps that are smaller than a micrometer (10^{-6} m). Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting composite geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Convert to Curve

To unite and convert geometry objects to single a curve object, on the **Geometry** toolbar, **Conversions** menu, click **Convert to Curve** (). You can also right-click the **Geometry** node to add this node from the **Conversions** submenu. Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** if you experience problems with the convert operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries or edges, and points) that the resulting curve object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Convert to Point

To unite and convert geometry objects to single a point object, on the **Geometry** toolbar, **Conversions** menu, click **Convert to Point** (). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** if you experience problems with the convert operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create a predefined selection for the points that the resulting geometric object consists of. This selection is available in all selection lists for points but does not appear as a separate selection node in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Convert to Solid

To unite and convert geometry objects to single a solid object, on the **Geometry** toolbar, **Conversions** menu, click **Convert to Solid** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** if you experience problems with the convert operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains and all or some of boundaries, edges, and points) that the resulting solid object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Convert to Surface

To unite and convert geometry objects to single a surface object, on the **Geometry** toolbar, **Conversions** menu, click **Convert to Surface** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** if you experience problems with the convert operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries, edges, and points) that the resulting surface object consists of. These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Copy

Use the **Copy** ( ) node to make a displaced copy of one or several geometry objects. This method creates a node in the model tree that contains a reference to other objects in the geometry sequence that are copied and keeps the objects linked (unlike a simple copy and paste function).

To create copies of geometry objects, on the **Geometry** toolbar click **Copy** ( ). You can also right-click the **Geometry** node to add this node from the **Transforms** submenu.

> To copy geometry features, you can also right-click the geometry feature in the model tree (for example, **Rectangle** or **Sphere**) and select **Copy** ( ). Then right-click the **Geometry** node and select **Paste** (for example, **Paste Rectangle** or **Paste Sphere**) ( ). It is also possible to copy and duplicate nodes corresponding to operation features such as the Union node.

Enter the properties of the copy operation using the following sections:

INPUT

Select the geometry objects that you want to copy in the Graphics window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Copy** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Clear the **Keep input objects** check box to remove the input objects.

DISPLACEMENT

Set the displacement in each direction by entering **x**, **y**, and **z** (not all fields are available in 1D and 2D); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To create several copies, enter a comma-separated or space-separated list of displacements in these fields, or click the **Range** button ( ) to use the **Range** dialog box for specifying a range of displacements for multiple copies.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

> - Copying, Pasting, and Duplicating Nodes
> - Copy and Paste Geometry Objects

Cross Section

In a **Work Plane** node's **Plane Geometry** sequence you can add a **Cross Section** node ( ). By default, this computes the cross section of all 3D objects generated by preceding nodes in the geometry sequence. You can also select specific 3D objects to intersect with the work plane. You can also add a new 2D or 2D axisymmetric **Component** and add the **Cross Section** node there. In that case you can select which **Work Plane** to use. For example, if you have a 3D geometry that is symmetric about an axis, you can add a work plane that contain the axis. In the axisymmetric 2D Component you then get the cross section and can use a 2D axisymmetric physics, which is computationally efficient compared to a full 3D Component. A **Cross Section** node can also be useful to extract a planar surface for modeling a thin flat 3D structure using shell elements, for example. To add a cross section, right-click a **Plane Geometry** node under a **Work Plane** node or a 2D **Geometry** node and select **Cross Section** ( ). Then enter the properties of the cross section using the following sections:

CROSS SECTION

If you add the **Cross Section** node to a 2D or 2D axisymmetric geometry, first select the work plane to use for the cross section from the **Work plane** list.

From the **Intersect** list, choose **All objects** (the default) to intersect all 3D geometry objects with the work plane, or choose **Selected objects** to intersect only the geometry objects that you add to the **Objects to intersect** list that appears. Click the **Active** button to toggle between turning ON and OFF the **Objects to intersect** selections.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the cross section operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cross-sectional geometry consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Deformed Configuration

To create a geometry from a (deformed) mesh, under **Results>Data Sets**, right-click a **Solution** data set and select **Remesh Deformed Configuration** ( ). The new geometry is added under **Meshes** as a **Deformed Configuration** node ( ). In the settings window of this node, the **Time** or **Parameter value** list controls which solution is used to generate the deformed configuration. If you change the time or parameter value, or if the solution itself has changed, update the deformed configuration by clicking the **Update** button. The meshes that belong to the deformed configuration appear as child nodes under the deformed configuration node. Thus, to remesh the deformed configuration, right-click such a mesh node and select **Build All**.

- Remeshing a Deformed Mesh
- Solution (data set)

Delete Entities

To delete geometry objects or geometric entities from objects, right-click a geometry and select **Delete Entities** ( ). Then enter the properties of the delete operation in the **Input** section. If you delete objects corresponding to primitive features these nodes disappear from the sequence. If you delete other objects or if you delete geometric entities a Delete Entities node appears in the sequence.

> If you want to delete all objects created by a feature, it is better to right-click the feature, and select **Delete** (✕) or **Disable** (⊘).

ENTITIES OR OBJECTS TO DELETE

From the **Geometric entity level** list, choose the level of the entities to delete: **Object**, **Domain**, **Boundary** (that is, faces in 3D and edges in 2D), **Edge** (3D only), or **Point**. Then select the objects or entities that you want to delete in the **Graphics** window or use the **Selection List** window. The objects appear in the **Selection** list when you have confirmed (locked) the selection in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Delete Entities** node, choose **Manual** to select objects or entities, or choose one of the selection nodes from the list next to **Selection**.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Difference

To subtract geometry objects from other geometry objects to make a set difference, on the **Geometry** toolbar, click **Difference** (⧉). You can also right-click the **Geometry** node to add this node from the **Boolean Operations** submenu. Then enter the properties of the difference operation using the following section:

INPUT

Activate the **Objects to add** list by clicking the **Active** button to toggle ON and OFF. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF. Then select the objects that you want to add in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Difference** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Objects to add**.

Activate the **Objects to subtract** list by clicking the **Active** button to toggle ON and OFF. Then select the objects that you want to subtract in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Difference** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Objects to subtract**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the difference operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Edit Object

Use the **Edit Object** (✎) feature to adjust the edges and vertices for a 2D geometry object or to add or delete edges and vertices in the object. To edit a 2D object using the settings window, in the Model Builder, right-click the Geometry node and select Edit Object from the context menu. To edit a 2D object using the Graphics window, select the object and click the **Edit Object** button. See Editing 2D Geometry Objects.

SELECT OBJECT

Select the **Input object** to edit in the **Graphics** window. When selected, a copy is made of the input object, and subsequent changes operate on this copy. Changes made in the **Input object** after the copy has been made has no effect on the **Edit Object** feature. For this reason, it is not possible to change the **Input object** after it has been selected.

Click the **Active** button to toggle between turning ON and OFF the **Input object** selections.

EDIT EDGES

Click the **Active** button to toggle between turning ON and OFF the **Edge** selections.

In the **Graphics** window, select and add the **Edge** number to edit (or use The Selection List Window). The parameters for the current edge are displayed in the table for **x (m)**, **y (m)**, (**xw (m)**, **yw (m)** in work planes)—where the **m** in parentheses indicates the current geometry length unit; in this case the default unit: meter—and **Weights**, and under the table for the **Degree** list and the **Start vertex** and **End vertex** fields.

If required, click **New** to create a new edge. The new edge is linear, not connected to any other edges or vertices and has both the start and the end coordinates set to 0. Click **Delete** to delete the current edge. Deleting an edge also deletes its adjacent vertices, if these vertices are not connected to other edges.

Click in the table cells to edit the **x (m)**, **y (m)**, (**xw (m)**, **yw (m)** in work planes) and **Weights** and modify the control points of the edge. If the **x** or **y** value for the first or last control point is modified, any adjacent edges and vertices are automatically updated with the same value.

Select the **Degree**—**Linear**, **Quadratic**, or **Cubic** to change the degree of the edge. When decreasing the degree, the control points are recalculated so as to approximate the old shape of the edge.

Under **Start vertex**, click the **Active** button to move the start point of the edge to a different vertex. In the **Graphics** window, select and add the **Edge** number. Under **End vertex**, click the **Active** button to move the end point of the edge to a different vertex. In the **Graphics** window, select and add the **Edge** number. If required, click **Disconnect** to disconnect the start or end vertex of an edge from the rest of the object. A new vertex is created with coordinates matching the start or end point of the edge. This vertex can be moved without affecting the other edges that were previously connected to this edge.

When the **Edit Object** node is the current node, you can visualize the edited object by observing the edge or vertex numbers displayed in the **Graphics** window next to the edges or vertices.

EDIT VERTICES

Click the **Active** button to toggle between turning ON and OFF the **Vertex** selections.

In the **Graphics** window, select and add the **Vertex** number (node number) to edit (or use The Selection List Window). The parameters for the current vertex are displayed in the **x** and **y** fields under **Coordinates**, and the vertex is highlighted in the **Graphics** window.

- Click **New** to create a new vertex; the coordinates for the new vertex are set to 0.
- Click **Delete** to delete the current vertex. Only isolated vertices can be deleted. For other vertices, the **Delete** button is disabled.
- Click **Snap to Closest** to delete the current vertex. Any edges connected to the deleted vertex are modified so that the start or end point is moved to the closest remaining vertex.

Under **Coordinates** edit the **x** and **y** fields as required.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Editing 2D Geometry Objects

Extrude

To extrude planar objects into 3D, on the **Geometry** toolbar, click **Extrude** (). You can also right-click the **Geometry** or a **Work Plane** feature to add this from the context menu. Then enter the properties of the extrude operation.

GENERAL

From the **Extrude from** list, select **Faces** to extrude planar faces from the 3D geometry. Select the faces that you want to extrude in the Graphics window. They appear in the **Input faces** list. All selected faces must lie in the same plane. Alternatively, from the **Extrude from** list, select **Work plane** to extrude objects from a work plane. In the **Work plane** list, select the work plane to extrude from. Select the objects that you want to extrude in the Graphics window. They appear in the **Input objects** list.

Select the **Unite with input objects** check box to unite the input objects with the extruded objects. Clear the **Unite with input objects** check box to keep the extruded objects separate from the input objects.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

DISTANCES FROM PLANE

Specify one or several distances in the table. These determine the height above the work plane or planar faces for the top of each layer. The **Reverse direction** check box reverses the direction of the extrusion.

If you extrude several layers, remove the interior boundaries by clearing the **Keep cross-sectional faces** check box.

The direction arrow that appears in the Graphics window indicates the length of each extrusion distance.

SCALES

For each layer, specify a length scaling factor for the top of the layer relative to the work plane object or planar faces.

DISPLACEMENTS

For each layer, specify a displacement vector for the top of the layer in the work plane's coordinate system, or the local coordinate system defined by the first face to extrude. The first face is the face with smallest face number in the geometry object that comes first in the geometry sequence.

TWIST ANGLES

For each layer, specify a rotation angle for the top of the layer around the work plane's normal vector or the normal vector of the first face to extrude. The first face is the face with smallest face number in the geometry object that comes first in the geometry sequence.

POLYGON RESOLUTION OF EDGES

This setting determines how accurately the edges in the extrusion direction are represented.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Fillet

To fillet corners in 2D geometry objects, on the **Geometry** toolbar click **Fillet** (). You can also right click the **Geometry** node and add this node from the context menu. Then enter the properties of the fillet operation using the following sections:

POINTS

Select the points (vertices) that you want to fillet in the **Graphics** window. They then appear in the **Vertices to fillet** list. If the geometry sequence includes user-defined selections above the **Fillet** node, choose **Manual** to select points, or choose one of the selection nodes from the list next to **Vertices to fillet**.

Click the **Active** button to toggle between turning ON and OFF the **Vertices to fillet** selections.

RADIUS

Enter the **Radius** of the circular fillet arc.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

If, Else If, Else, End If

To construct an If statement, on the **Geometry** toolbar, **Programming** menu, select **If + End If** (you can also do this by right-clicking the **Geometry** node in the model tree and opening the **Programming** submenu). This adds an **If** node () and an **End If** node () after the current node. You can optionally add **Else If** () nodes and an **Else** node

() in a similar way. To add these nodes at an arbitrary position in a geometry sequence you can right-click a geometry feature node and select **If**, **Else If**, **Else**, or **End If** on the **Add Before** or **Add After** submenu. This adds the selected type of programming feature before of after the selected node, without building the preceding feature node.

An If statement has the following structure:

```
If
  <branch1>
Else If
  <branch2>
Else If
  <branch3>
...
Else
  <last branch>
End If
```

where the **Else If** and **Else** nodes are optional. There can be an arbitrary number of geometry features in each branch, and there can be an arbitrary number of **Else If** nodes. The **Else** node must appear after all **Else If** nodes and before the **End If** node.

IF AND ELSE IF

The **If** and **Else If** nodes' settings windows have a **Condition** field, which contains a logical condition in terms of parameters (for example, `a+b>0`, where `a` and `b` are defined as parameters for the geometry sequence). In general, the condition is true if it evaluates to a nonzero value. When building the geometry sequence, the program builds the features in the first branch that has a true condition and treats the other branches as disabled. If none of the conditions are true, the program builds the **Else** branch.

If you select a feature in a branch and click **Build Selected**, the software pretends that the chosen branch has a true condition and that all other branches have false conditions. You can use this behavior to try out the different branches without having to change the parameters. If statements can be nested.

To define selections that have different definitions in different branches of an If statement, you can use cumulative selections (see Cumulative Selections).

Import

To import geometry objects from a file or from another geometry, on the Geometry toolbar click **Import** (). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the import feature in the **Import** section of the settings window for the **Import** node.

Importing a Virtual Geometry

If you import a file or sequence containing a virtual geometry, only the underlying "real geometry" is imported.

IMPORT

In the **Geometry import** list choose the type of data to import—**Any importable file**, **Geometry sequence**, and **COMSOL Multiphysics file** are always available. In addition, you can choose **DXF file** in 2D and **STL/VRML file** in 3D.

> If you have license for the CAD Import Module, you also have **3D CAD file** in 3D (see the *CAD Import Module User's Guide* for more information about supported CAD file formats).

The alternative import formats, **ECAD file (GDS/NETEX-G)** and **ECAD file (ODB++)** are available in 2D and 3D with a license for the ECAD Import Module. See the *ECAD Import Module User's Guide* or go to http://www.comsol.com/ecad-import-module/ for more information.

Combining several objects where at least one is imported from an STL/VRML file is not supported.

For file import, specify the file name in the **Filename** field or click the **Browse** button. For import from another geometry, select the geometry sequence from the **Source** list below. For DXF, STL/VRML, and 3D CAD import, you can change a number of properties when you have selected the file type. To import the file, click the **Import** button ( ). If you have changed some property, the software automatically re-imports the file when you click a build button.

Properties for DXF import

The repair tolerance specifies the largest distance between the end points of curves allowed in the imported geometry. You can specify this tolerance as an import option.

In the **Layer selection** list, select the layers to import.

Under **Import options**, select **Form solids** to unite and convert all objects in each layer to a solid object, select **Knit curves** to unite and convert all objects in each layer to a curve object, or select **Do not knit** to do nothing.

If the **Repair imported objects** check box is selected, enter a **Relative repair tolerance**. To create a geometry for mesh generation and finite element analysis, COMSOL requires a high degree of accuracy within the CAD drawing. Sometimes DXF geometries contain small gaps and exceedingly short edges that make it impossible to create a valid 2D solid or a valid mesh. COMSOL provides repair tolerance settings to remove short edges and close small gaps during DXF file import. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the imported objects (the default value is 10^{-5}). Geometric entities that have a distance less than the absolute repair tolerance are merged.

Properties for STL/VRML import

You can import 3D surface meshes into COMSOL with STL and VRML 1.0, two formats for describing solids as surface meshes:

- The STL format describes the boundary of a solid as a triangular mesh. STL files are either ASCII or binary, and COMSOL can import both ASCII and binary STL files. It is advisable to use binary files if possible to avoid a loss of precision.
- The VRML format describes scenes (collections of independent objects that reside within a single "frame" or logical space), and it can describe the geometry in a scene as a triangular mesh. COMSOL ignores all other information in a VRML file. COMSOL only imports VRML 1.0 files.

The imported objects are represented using COMSOL's geometry kernel, so you cannot apply CAD defeaturing or repair on them.

These two formats do not contain any information about to what face each triangle belongs. Instead, COMSOL uses heuristics to partition the triangles into faces. Usually the partition is reasonable for geometries containing sharp angles between faces, but the partitioning can appear arbitrary for curved geometries such as spheres and tori.

The mesh triangles are automatically partitioned into faces so that you get edges where neighboring triangles make large a large angle. To control this algorithm, change **Face partitioning** to **Manual**. The angle between any two triangles in the same face is less than the **Maximum angle within face** value. The angle between neighboring triangles in the same face is less than the **Maximum face neighbor angle** value.

If the **Detect planar faces** check box is selected, (approximately) planar faces are detected. A planar face has an area larger than the total area of all faces times the value **Minimum relative area**. The angle between neighboring triangles in the same planar face is less than the value in the **Maximum neighbor angle** field.

A small face is removed if its area is less than the mean area of all faces times the value in the **Removal of small faces** field.

Advanced STL/VRML import parameters

If you select **Manual** from the **Advanced parameters** list, some additional controls for detection of extruded faces and faces with constant curvature become visible.

In an extruded face all triangles are approximately orthogonal to the extruded plane (work plane). The angle between such a triangle's normal and the extruded plane is less than the value in the **Maximum angle to extruded plane** field. An extruded face has an area larger than the total area of all faces times the value in the **Detect extruded faces** field.

In a face with constant curvature, the relative deviation of the curvature at neighboring triangles is at most the value in the **Maximum curvature deviation in face** field. A face with constant curvature has an area larger than the total area of all faces times the value in the **Detect constant curvature** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. For ECAD import, if more than one layer is imported, the **Create selections** check box additionally creates individual selections for each imported layer. For other types of import, if more than one object is imported, the **Create selections** check box creates individual selections for each imported object. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

See the *ECAD Import Module User's Guide* or go to http://www.comsol.com/ecad-import-module/ for more information.

Eigenmodes of a Room: model library path **COMSOL_Multiphysics/Acoustics/eigenmodes_of_room**

Intersection

To create the intersection of geometry objects, on the **Geometry** toolbar, click **Intersection** (). You can also right-click the **Geometry** node to add this node from the **Boolean Operations** submenu. Then enter the properties of the intersection operation using the following section:

INPUT

Select the geometry objects that you want to intersect in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Intersection** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

To create a geometry object without interior boundaries, clear the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the intersection operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Mirror

To mirror (reflect) geometry objects in a plane (3D), a line (2D), or a point (1D), on the **Geometry** toolbar, click **Mirror** ( ). You can also right-click the **Geometry** node to add this node from the **Transforms** submenu. Then enter the properties of the mirror operation using the following sections:

INPUT

Select the geometry objects that you want to reflect in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Mirror** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

POINT ON PLANE/LINE OF REFLECTION

	Specify a point to be fixed during reflection by entering **x**, **y**, and **z**.

NORMAL VECTOR TO PLANE/LINE OF REFLECTION

	Specify a vector in the direction to reflect by entering **x**, **y**, and **z**.

POINT OF REFLECTION

—	For a 1D model, specify the coordinate of the point of reflection in the **x** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Move

To move (translate) geometry objects, on the Geometry toolbar click **Move** ( ). You can also right-click the **Geometry** node to add this node from the **Transforms** submenu. Then enter the properties of the move operation using the following sections:

INPUT

Select the geometry objects that you want to move in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Move** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

DISPLACEMENT

Set the displacement in each direction by entering **x**, **y**, and **z** (not all fields are available in 1D and 2D geometries); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To create several copies, enter a comma-separated or space-separated list of displacements in these fields, or click the **Range** button ( ) to use the **Range** dialog box for specifying a range of displacements for moving multiple copies.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Partition

The **Partition** node ( ) provides a way to partition geometry objects as a Boolean operation. Partitioning a geometry objects can be useful to create separate domains or to introduce an interior boundary, for example. Using the **Partition** node you can partition a target object using a set of *tool objects* (geometry objects that are only used to partition—or tool—other geometry objects) or using an (infinite) plane defined by a **Work Plane** node (you do not need to draw anything in the work plane). To add it to a model, on the **Geometry** toolbar, click **Partition**. You can also right-click the **Geometry** or a **Work Plane** feature to add this from the **Boolean Operations** submenu. Then enter the properties of the partitioning operation using the following sections:

PARTITION

In the **Objects to partition** list, add the geometry objects that you want to apply a partition operation on. Click the **Active** button to toggle between turning ON and OFF the **Objects to partition** selections.

From the **Partition with** list, select **Objects** (the default) to partition using the geometry objects that you add to the **Tool objects** list below, or select **Work plane** to partition using any of the added work planes.

- If you select **Objects**, add the geometry objects that you want to use as tool object to the **Tool objects** list. Click the **Active** button to toggle between turning ON and OFF the **Tool objects** selections. Those geometry objects are only used to partition the geometry objects in the **Objects to partition** list and are not included in the finalized geometry used for defining materials and physics.
- If you select **Work plane**, select from the available work planes in the **Work plane** list. Click the **Go to Source** button ( ) to move to the **Work Plane** node for the selected work plane.

Select the **Keep input objects** check box to use the selected geometry objects to for further geometry operations.

Adjust the **Relative repair tolerance** (default: 10^{-6}) if you experience problems with the partition operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Revolve

To revolve planar objects into 3D, right-click a geometry or a work plane node, and select **Revolve** ( ). Then enter the properties of the revolve operation.

GENERAL

From the **Revolve from** list, select **Faces** to revolve planar faces from the 3D geometry. Select the faces that you want to revolve in the Graphics window. These appear in the **Input faces** list. All selected faces must lie in the same plane.

Alternatively, from the **Revolve from** list, select **Work plane** to revolve objects from a work plane. In the **Work plane** list, select the work plane to revolve from. Select the objects that you want to revolve in the Graphics window. They appear in the **Input objects** list.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** or **Input faces** selections.

Select the **Unite with input objects** check box to unite the input objects with the revolved objects. Clear the **Unite with input objects** check box to keep the revolved objects separate from the input objects.

REVOLUTION ANGLES

Click the **Angles** button to specify the start angle (default: 0 degrees) and end angle (default value: 360 degrees; that is, a full revolution) of the revolution. If you click the **Full revolution** button you get a full revolution. This selection also enables the **Keep original faces** check box that controls if the original faces are kept in the revolved 3D geometry. Keeping these faces is necessary if you want to create a swept mesh. By default, COMSOL keeps such faces.

REVOLUTION AXIS

Select **2D** in the **Axis type** list to specify the revolution axis in the local coordinate system. When revolving work plane objects, the local coordinate system is defined by the work plane's coordinate system. When revolving planar faces, the local coordinate system is defined by the face with the smallest face number in the first geometry object in the geometry sequence. Select **3D** in the **Axis type** list to specify the revolution axis in the 3D coordinate system.

POINT ON THE REVOLUTION AXIS

Specify a point on the revolution axis in the local coordinate system in the **xw** and **yw** fields. Alternatively, if **Axis type** is **3D**, specify a point on the revolution axis in the 3D coordinate system in the **x**, **y**, and **z** fields.

DIRECTION OF REVOLUTION AXIS

Specify a direction vector for the revolution axis in the local coordinate system in the **xw** and **yw** fields. Alternatively, if **Axis type** is **3D**, specify a direction vector for the revolution axis in the 3D coordinate system in the **x**, **y**, and **z** fields.

POLYGON RESOLUTION OF EDGES

This setting determines how accurately the edges in the revolution direction are represented.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Rotate

To rotate geometry objects, on the **Geometry** toolbar, click **Rotate** (↺). You can also right-click the **Geometry** or a **Work Plane** feature to add this from the **Transforms** submenu. You can create one or multiple rotated copies with varying rotation angle. Then enter the properties of the rotate operation:

INPUT

Select the geometry objects that you want to rotate in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Rotate** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

ROTATION ANGLE

Specify the rotational angle in the **Rotation** field (default: 0 degrees). To get several rotated objects, enter a list of angles separated with commas or spaces or using the `range` function. Click the **Range** button ( ) to define a range of angles using the **Range** dialog box. For example, `range(0,45,315)` creates eight objects, one at the original position and seven rotated copies at 45 degrees distance around a full 360 degrees circle.

CENTER OF ROTATION

Enter the center of the rotation in the **x** and **y** (for 2D) and **r** and **z** (in 2D axial symmetry) and 2D axisymmetric models for **xw** and **yw** in work planes fields.

POINT ON AXIS OF ROTATION

Enter a point on the rotation axis in the **x**, **y**, and **z** fields.

AXIS OF ROTATION

From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Scale

To scale geometry objects, on the **Geometry** toolbar, click **Scale** ( ). You can also right-click the **Geometry** or a **Work Plane** feature to add this from the **Transforms** submenu.Then enter the properties of the scale operation using the following sections:

INPUT

Select the geometry objects that you want to scale in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Scale** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

SCALE FACTOR

By default, you get an isotropic scaling. Specify the scaling factor in the **Factor** field.

To get an anisotropic scaling, change **Scaling** to **Anisotropic**, and specify separate scale factors for the coordinate directions in the **x**, **y**, and (3D only) **z** fields; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes.

CENTER OF SCALING

Specify the center point of the scaling by specifying **x**, **y**, and(3D only) **z**; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. This is the point that stays fixed during the scaling (that is, the point that the scaled geometry objects approach when the scale factor goes to zero).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Split

The **Split** ( ) operation splits an object into its entities:

- A solid splits into solids corresponding to its domains.
- A surface object splits into surface objects corresponding to its faces.
- A curve object splits into curve objects corresponding to its edges.
- A point object splits into point objects corresponding to its vertices.
- A general (mixed) object splits into solids (corresponding to the domains), surface objects (corresponding to faces not adjacent to a domain), curve objects (corresponding to edges not adjacent to a face or domain), and point objects (corresponding to vertices not adjacent to an edge, face, or domain).

To split geometry objects into their entities, on the **Geometry** toolbar, from the **Conversions** menu, select **Split** ( ). Then enter the properties of the split operation in the **Input** section.

INPUT

Select the geometry objects to split on in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Split** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Subsequence Call

Use a **Subsequence Call** feature node to call and run a geometry subsequence as part of the model component's geometry sequence. The geometry subsequence can be a global subsequence defined under **Global Definitions** or a locally-defined subsequence. A default Mirror node is added but is removed if you change the call to a global subsequence. To call a geometry subsequence, right-click a geometry and select **Programming>Subsequence Call**. This adds a **Subsequence Call** node ( ), which builds an instance of the subsequence with new values of its arguments. Enter the properties of the feature using the following sections:

SUBSEQUENCE

Choose the subsequence to call in the **Subsequence** list. You can choose among the subsequences defined under **Global Definitions**. You can also choose **Local subsequence** (the default if no other subsequences exist), which means that the subsequence is defined by the **Local Subsequence** node ( ) beneath the **Subsequence Call** node.

ARGUMENTS

The **Name** column of the table contains the arguments required by the subsequence. In the **Expression** column, enter the corresponding expressions. The expressions can contain parameters defined under **Global Definitions**. The values of these expressions appear in the **Value** column. The **Description** column contains the descriptions given by the subsequence.

You can import or load data in files from a spreadsheet program, for example, with the **Load from file** button ( ) and the **Load from File** dialog box that appears. Data must be separated by spaces or tabs. If the license includes LiveLink™ *for* Excel® you can also load arguments from a Microsoft Excel Workbook spreadsheet.

POSITION AND ORIENTATION OF OUTPUT

Use this section to specify the translation and rotation to apply to the output objects (not in 1D).

In 3D, this transformation is done in two steps:

- The first step is to transform the output objects so that a chosen coordinate system from the subsequence matches a given coordinate system defined by some feature preceding the Subsequence Call feature. Both coordinate systems are defined as the local coordinate system of a work plane. By default, the first step does nothing.
- The second step is to transform the output objects using a displacement and a rotation.

In 2D, only the second step of the transformation is available.

Coordinate system in subsequence

Choose the coordinate system from the subsequence in the **Work plane in subsequence** list. You can choose among all work plane features in the subsequence. You can also choose **xy-plane**, which corresponds to the global coordinate system (this is the default).

Coordinate system to match

To choose the coordinate system to match, first make a choice in the **Take work plane from** list. The default is **This sequence**, which means that you can select a work plane feature from this sequence. You can also choose another **Subsequence Call**, which means that you take the work plane from the other **Subsequence Call** node's subsequence. In both cases, you select the work plane to match in the **Work plane** list. The default is **xy-plane**, which corresponds to the global coordinate system. The work plane to match is visualized in the graphics.

Displacement

Enter values or expressions for the **xw**, **yw**, and **zw** coordinates (SI unit: m) to add a displacement vector relative to the coordinate system to match.

Rotation

Select an **Axis type**—**xw-axis**, **yw-axis**, **zw-axis** (the default), **Cartesian**, or **Spherical**. For any choice, enter a **Rotation angle** (SI unit: degrees; default value: 0) to rotate the orientation of the output relative to the coordinate system to match.

If **Cartesian** is selected, enter Cartesian coordinates values for **xw**, **yw**, and **zw** (default values 0, 0, and 1 corresponding to the global *zw*-axis) to specify the axis vector. If **Spherical** is selected, specify the axis vector using spherical angles **theta** and **phi** (SI unit: degrees; default 0).

Finally, specify a **Rotation angle** about the chosen axis (default: 0).

In 2D, only the second step of the transformation is available. You can specify a displacement vector and a rotation angle.

OBJECT SELECTIONS

When you have built the Subsequence Call feature, the table shows the names of the output object selections that are defined by the subsequence. When you click a row in the table, the corresponding selection is highlighted in the graphics. These selections are available for use as input to following geometry features. If you want a selection to contribute to a cumulative selection, choose that cumulative selection from the corresponding list in the **Contribute to** column. To contribute to a new cumulative selection, click the selection in the table and then the **New Cumulative Selection** button. This opens a dialog box where you can specify the name of the cumulative selection.

DOMAIN/BOUNDARY/EDGE/POINT SELECTIONS

These sections are similar to the **Object Selections** section. These selections are also available for use in, for example, the mesh and physics settings.

Sweep

Select **Sweep** ( ) from the **Geometry** toolbar to sweep one or several faces along a curve. Then enter the properties of the sweep operation using the following sections:

CROSS SECTION

Select the faces you want to sweep in the **Graphics** window. The faces appear in the **Faces to sweep** list. Click the **Active** button to toggle between turning ON and OFF the **Faces to sweep** selections.

Select the **Create cross-sectional faces** check box (active by default) to make the sweep operation create cross-sectional faces between the sweep sections. Such cross-sectional faces can be useful, for example, for a swept mesh where you want to specify the mesh distribution for each section of the sweep.

SPINE CURVE

Select the edges you want to sweep along in the **Graphics** window. More than one edge can be selected, but the selected edges must form a nonclosed connected chain. The edges appear in the **Edges to follow** list. Click the **Active** button to toggle between turning ON and OFF the **Edges to follow** selections.

Select the **Reverse direction** check box to sweep in the negative edge direction.

KEEP INPUT

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Select the **Include all inputs in finalize operation** check box to force the objects in the **Face to sweep** and **Edges to follow** lists to be included in the Form Union/Assembly operation. If the **Include all inputs in finalize operation** check box is not selected, these objects are not included in the Form Union/Assembly operation if the **Face to sweep** list or **Edges to follow** list contains all faces or edges in the objects.

MOTION OF CROSS SECTION

This section contains a number of properties that determine how the face is transformed when swept along the spine curve.

A curve parameter name can be defined in the **Parameter name** field. Use this parameter in the expressions defining scale factor and twist angle.

The **Scale factor** field controls the size of the cross section face when swept along the spine curve.

The **Twist angle** field controls the rotation angle of the cross section face about the spine curve.

By default, twist compensation is active and prevents the twisting that would otherwise occur due to nonzero torsion for nonplanar curves. Clear the **Twist compensation** check box to turn off this compensation. When **Twist compensation** is active, it behaves as if a term was added to the **Twist angle** with a magnitude matching the integral of the torsion of the curve. This makes the edges in the sweep direction locally parallel to the spine curve. For a noncircular cross section, twist compensation also affects the shape of the generated object.

From the **Face-spine alignment** list, select an option to align the cross section to the spine curve:

- Select **No adjustment** (the default) to sweep the face starting from its original position. Using this setting, it is possible to create sweeps where the face is not perpendicular to the spine curve, and where the face does not contain that start point of the spine curve.
- Select **Adjust spine** to adjust the spine curve so that it starts on the face to sweep and so that it is parallel to the face normal at the point where it touches the face. The first part of the spine curve is replaced by a cubic Bézier curve, with the length of the replaced part, measured in parameter values, controlled by the value in the **Adjustment parameter length** field.
- Select **Move face** to move the face to the start of the spine curve and orient the face perpendicularly to the spine curve. This setting is only allowed when the face is located in a work plane, and the movement is such that the work plane origin coincides with the spine curve.

ADVANCED SETTINGS

From the **Geometry representation** list, select **Spline** (the default) to represent the swept object using splines, or **Bézier**, to represent the swept object using Bézier curves. The difference is that using Bézier curves, the intersections between the surfaces that form the swept object are visible edges, whereas they are hidden when using splines.

The values in the **Relative tolerance** field is a relative tolerance that controls the accuracy of the geometric representation of the swept object. The geometric representation is an approximation, which is necessary because it is not possible to exactly represent a swept object using NURBS (nonuniform rational basis splines). The default value is 10^{-4} (0.01%).

Internally, the software represents the swept object by B-spline curves and surfaces, which are computed to approximate the mathematical definition of the swept surface. The number of knot points in the splines increases automatically until the approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field (default value: 1000).

If more than one edge is selected in the **Edges to follow** list, the **Direction-defining edge** controls which edge is used to define the positive sweep direction. The **Direction-defining edge** is automatically set when the first edge is added to the **Edges to follow** list, so usually it does not have to be changed manually.

If the expressions for scale or twist contain user-defined functions, changes in those functions do not automatically cause the sweep feature to be rebuilt. To rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Tangent

To create a tangent (that is, a line segment tangent to a specified edge), on the **Geometry** toolbar, click **Tangent** ( ). You can also right-click the **Geometry** node to add this node from the context menu. Then enter the properties of the tangent.

TANGENT

Select the edge you want to tangent in the **Graphics** window. Only one edge can be selected. The edge appears in the **Edge to tangent** list.

Click the **Active** button to toggle between turning ON and OFF the **Edge to tangent** selections.

Parameter start guess is a number between 0 and 1 that specifies where on the edge the expected point of tangency is located. The tangent returned is the first tangent found starting the search from the start guess.

Type of tangent specifies how the end point of the tangent is specified. There are three options: **Edge–edge**, **Edge–point**, and **Edge–coordinate**.

When the type is edge–edge, you use the **Graphics** window to select a second edge to tangent. The edge appears in the **Second edge to tangent** box. The line segment created is tangent to both edges. Use **Parameter start guess for second edge** to specify the start guess for the second edge, it is a number between 0 and 1. Click the **Active** button to toggle between turning ON and OFF the **Second edge to tangent** selections.

When the type is edge–point, you use the **Graphics** window to select a point in the geometry as tangent's endpoint. The point appears in the **Point** list.

When the type is edge–coordinate, manually specify the coordinates of the tangent's endpoint.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Union

To create the union of geometry objects, on the **Geometry** toolbar, click **Union** ( ). You can also right-click the **Geometry** node to add this node from the **Boolean Operations** submenu. Then enter the properties of the union operation.

INPUT

Select the geometry object that you want to unite in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Union** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Click the **Active** button to toggle between turning ON and OFF the **Input objects** selections.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default: 10^{-6}) if you experience problems with the union operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

Work Plane

To create a work plane for defining 2D objects in 3D (for example, extruding a 2D object into a 3D object) or for defining the placement of the resulting objects of a Subsequence Call feature, right-click a 3D **Geometry** node and select **Work Plane** ( ). Then enter the properties defining the location of the work plane in the following sections of its settings window.

Using Work Planes

PLANE DEFINITION

Plane Type

The layout of the **Work Plane** section depends on the selection in the **Plane type** list, where you select how to define the work plane. Choose one of the following types:

- Quick (the default)
- Face parallel
- Edge parallel
- Edge angle
- Circle perpendicular
- Vertices
- Coordinates
- Transformed

Quick

In the **Plane** list, select one of the global coordinate planes *xy*, *yz*, *zx*, *yx*, *zy*, or *xz*, denoting the first and second axes of the work plane's local coordinate system. Specify an offset using on of the following settings in the **Offset type** list:

- **Distance** (the default) to define the distance from the coordinate plane in the third axis' direction using the **z-coordinate**, **x-coordinate**, or **y-coordinate** field (default value: 0; that is, no offset).
- **Through vertex** to define the position of the work plane in the third direction by selecting a vertex. The work plane's position then contains that vertex. Click the **Active** button to toggle between turning ON and OFF the **Offset vertex** selections.

Face parallel

Select a planar face in the **Graphics** window that is parallel to the work plane you want to create. The **Planar face** list shows the selected face. Click the **Active** button to toggle between turning ON and OFF the **Planar face** selections. Specify an offset using on of the following settings in the **Offset type** list:

- **Distance** (the default) to define the distance in the **Offset in normal direction** field. You then offset the work plane along the normal of the planar face. By default, the work plane's normal is the outward normal of the face in the **Planar face** list.The default value: 0; that is, no offset.
- **Through vertex** to define the position of the work plane in the third direction by selecting a vertex. The work plane's position then includes the position of that vertex. Click the **Active** button to toggle between turning ON and OFF the **Offset vertex** selections.

To reverse the direction of the *z*-axis of the work plane's coordinate system, select the **Reverse normal direction** check box. This also swaps the coordinate axes in the work plane to preserve the positive orientation of the local coordinate system.

Edge parallel

Select a planar edge (that is not straight) in the **Graphics** window that is parallel to the work plane you want to create. The **Planar edge** list shows the selected edge. Click the **Active** button to toggle between turning ON and OFF the **Planar edge** selections.

Specify an offset using on of the following settings in the **Offset type** list:

- **Distance** (the default) to define the distance in the **Offset in normal direction** field. You then offset the work plane along the normal of the plane containing the edge. The default value: 0; that is, no offset.
- **Through vertex** to define the position of the work plane in the third direction by selecting a vertex. The work plane's position then includes the position of that vertex. Click the **Active** button to toggle between turning ON and OFF the **Offset vertex** selections.

To reverse the direction of the *z*-axis of the work plane's coordinate system, select the **Reverse normal direction** check box. This also swaps the coordinate axes in the work plane to preserve the positive orientation of the local coordinate system.

Edge angle

Activate the **Straight edge** list by first selecting its **Active** button and then selecting a straight edge in the **Graphics** window. Similarly, activate the **Face adjacent to edge** list by first clicking its **Active** button and then selecting an adjacent face in the **Graphics** window. Also, specify a value in the **Angle between face and work plane** field (in degrees; the default value is 0). This results in a work plane through the given edge that makes the specified angle with the adjacent face.

By default, the origin of the local coordinate system coincides with the edge's start vertex, and the direction of the local *x*-axis coincides with the direction of the edge. If you select the **Reverse normal direction** check box, the origin instead is at the end vertex, and the normal direction of the local *x*-axis is reversed.

Circle perpendicular

Activate the **Circular edge** list by clicking its **Active** button. Then select a circular edge in the **Graphics** window. This results in a work plane perpendicular to the given circular edge. Use the **Point on plane** list to choose a vertex that the plane should go through:

- The edge's **Start vertex** (the default)
- The edge's **End vertex**.
- Some **Other verte**x, which you can choose from the **Vertex** list that opens.

You can then rotate the work plane around the normal direction of the circle's plane by specifying an **Angle offset** (in degrees; default value: 0 degrees). You can also reverse the work plane's normal direction using the **Reverse normal direction** check box.

The origin of the local coordinate system is at the circle's center. The local x-axis goes through the circle. Thus, if the geometry is rotationally symmetric, the symmetry axis coincides with the local y-axis. You can use this type of work plane together with a **Cross Section** node to get a 2D axisymmetric geometry corresponding to a rotationally symmetric 3D geometry.

Vertices

In each of the lists **First vertex**, **Second vertex**, and **Third vertex**, select a vertex by first clicking the corresponding **Active** button and then selecting a vertex in the **Graphics** window. This creates a work plane parallel to a plane through the three vertices.

Specify an offset in the **Offset in normal direction** field (default value: 0; that is, no offset). The origin of the local coordinate system is located above the first vertex, and the vector to the second vertex becomes the local x-axis. Reverse the directions of the local z-axis and y-axis by selecting the **Reverse normal direction** check box.

Coordinates

This choice creates a work plane through the three points with the given coordinates. The origin of the local coordinate system coincides with **Point 1**. The x-axis of the local coordinate system is in the direction of the vector from **Point 1** to **Point 2**. The positive direction of the y-axis is determined by the condition that the vector from **Point 1** to **Point 3** has a positive y-component.

Transformed

Use the work plane type to create a work plane using a transformation of another work plane. From the **Take work plane from** list, select **This sequence** (the default) to use a work plane earlier in the same geometry sequence, or choose a geometry subsequence that this geometry sequence calls earlier in the sequence. From the **Work plane to transform** list, select an available work plane (for example, **Work Plane 1 {wp1}**) from the selected geometry sequence or select **xy-plane** (the default).

Under **Displacement**, enter the desired displacements in the work plane's x-, y-, and z-directions in the **xw**, **yw**, and **zw** fields, respectively. The defaults are 0 (that is, no displacement).

Under **Rotation**, choose an axis for the rotation from the **Axis type** list:

- **xw-axis**, **yw-axis**, or **zw-axis** (the default) to use the local x-, y-, or z-axis.
- **Cartesian** to define an axis using local Cartesian coordinates in the **xw**, **yw**, and **zw** fields. The default values are 0, 0, and 1, respectively (that is, the same as choosing **zw-axis**).
- **Spherical** to define an axis using spherical angles (in degrees), which you enter in the **theta** and **phi** fields (default value for both angles: 0).

In the **Rotation angle** field enter the rotation angle (in degrees) about the specified axis (default: 0).

LOCAL COORDINATE SYSTEM

In this section you specify the local coordinate system in the work plane for most work plane types.

In the **Quick** work plane type:

- In the **Origin** list, choose the location of the origin of the work plane's coordinate system: **Global** (the default) or **Vertex projection**. In the latter case, also pick a vertex for the origin, which you add to the list under **Vertex for origin**.
- In the **Local x-axis** list, choose how to define the local x-axis: **Natural** (the default), which means that the local x-axis corresponds to the first direction in the plane; for example, the y direction for a yz-plane. Alternatively, choose **Through vertex projection** to define the local x-axis through a vertex projection. Then choose a vertex for the local x-axis that you add to the **Vertex for axis** list.

In the **Face parallel** plane type:

- In the **Origin** list, choose the location of the origin of the work plane's coordinate system: **Center of face** (the default), **Bounding box corner**, or **Vertex projection**. In the last case, choose a vertex for defining the origin that you add to the **Vertex for origin** list.
- In the **Local x-axis** list, choose how to define the local x-axis: **First parameter direction** (the default) or **Second parameter direction**, which are the local parameter directions of the face (represented by the variables `s1` and `s2`, respectively). Alternatively, choose **Through vertex projection** to define the local x-axis through a vertex projection. Then choose a vertex for the local x-axis that you add to the **Vertex for axis** list.

In the **Edge parallel** plane type:

- In the **Origin** list, choose the location of the origin of the work plane's coordinate system: The **Start vertex** (the default) or **End vertex** of the edge, or **Vertex projection**. In the last case, choose a vertex for defining the origin that you add to the **Vertex for origin** list.
- In the **Local x-axis** list, choose how to define the local x-axis: **Tangent direction** (the default), which means that the local x-axis follows the direction of the edge's tangent. Alternatively, choose **Through vertex projection** to define the local x-axis through a vertex projection. Then choose a vertex for the local x-axis that you add to the **Vertex for axis** list.

In the **Quick**, **Face parallel**, **Edge parallel**, **Edge angle**, **Circle perpendicular**, and **Vertices** work plane types:

- Enter displacements within the plane in the **xw-displacement** and **yw-displacement** fields if you want to move the origin of the local coordinate system. The defaults are 0.
- Enter a rotation angle in the **Rotation** field if you want to rotate the local coordinate system. The default is 0 degrees; that is, no rotation.

UNITE OBJECTS

By default, COMSOL unites all objects in the 2D work plane, which can make it easier to, for example, extrude the 2D geometry into a 3D geometry object.

Clear the **Unite objects** check box if you do not want to unite the separate 2D geometry objects in the work plane. If the check box is selected, COMSOL uses the tolerance given in the **Relative repair tolerance** field (default value: 10^{-6}) when uniting the geometry objects.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of boundaries, edges, and points) that the 2D geometric objects in the work plane consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree. If you want to make the selections contribute to a cumulative selection, select a cumulative selection from the **Contribute to** list, or click the **New** button to create a new cumulative selection (see Cumulative Selections).

DRAWING IN THE WORK PLANE

To show the work plane, click the **Show Work Plane** button ( ) in the settings window's toolbar, or click the **Plane Geometry** node that appears under the **Work Plane** node. To create 2D objects in the work plane, right-click the **Plane Geometry** node and create nodes like in a 2D geometry.

EMBEDDING THE WORK PLANE IN THE 3D GEOMETRY

To embed the 2D work plane geometry in the 3D geometry, build the Work Plane feature by either right-clicking the **Work Plane** node and choosing **Build Selected** or selecting the **Work Plane** node and then clicking **Build Selected** or **Build All Objects**.

Using Work Planes

CREATING 3D GEOMETRIES FROM 2D WORK PLANES AND 3D FACES

In addition to creating 3D geometries directly using 3D geometric primitives, it is also possible to form 3D geometries based on 2D sections (2D geometries) created in work planes or faces in the existing 3D geometry. A *work plane* is a 2D plane oriented anywhere in the 3D space. Quick options make it easy to create a work plane that is parallel to any of the main Cartesian planes or to a face or edge in an existing 3D geometry.

There are several methods to create 3D solid objects from 2D sections or faces. In addition, you can use a 2D section as an "embedded" surface in the 3D geometry.

Drawing on a work plane works just as drawing in 2D. The 3D work plane adapts its size to the drawn geometry.

See Electric Sensor (model library path **COMSOL_Multiphysics/Electromagnetics/electric_sensor**). Use the model instructions to practice building the geometry, which includes **Work Plane**, **Rectangle**, **Ellipse**, **Union**, **Compose**, **Extrude**, and **Block** features.

DRAWING ON A 2D WORK PLANE IN 3D

When using a **Work Plane** ( ) node to define 2D objects in 3D (for example, to extrude into a 3D object), the 3D projection settings enable you to draw on the work plane in 3D. These instructions provide an example. When the **Draw on work plane in 3D** check box is selected in the settings window for the **Plane Geometry** node ( ) under the **Work Plane** node, two additional buttons are available in the **Graphics** window—the **Align with work plane** button ( ) and the **Work plane clipping** button ( ). The standard 2D geometry draw toolbar is also available for use..

Some computer graphic cards cannot run the work plane rendering. In these cases, the work plane is rendered as a blue plane. It is possible to go to the 3D work plane but not to draw on the plane.

If you prefer to draw on the work plane in 3D, you can change the default. Open The Preferences Dialog Box, click **Geometry** and then click to select the **View work plane geometry in 3D** check box.

1 Add a 3D Component and then add a geometry (for example, a **Sphere**).

2 Right-click to add a **Work Plane** node under the main **Geometry** node.

3. Under **Work Plane**, click the **Plane Geometry** node.
4. Under **Visualization**, select the **View work plane geometry in 3D** check box. See Figure 7-3 and Figure 7-4 to see what happens when the check box is selected.
5. The geometry displays in the **Graphics** window. See Figure 7-4 for an example.
    - Click the **Align with work plane** button ( ) to rotate and move the camera to see the work plane from the top down.
    - Click to toggle the **Work plane clipping** button ( ) on and off. When on, use it to cut away all geometries above the work plane and make it easier to draw when objects are overlapping within the work plane. The clipping is not done when looking at the work plane from the side.
6. Under **In-plane visualization of 3D geometry,** specify how to visualize 3D objects in the work plane (as blue curves and points) by selecting one or more of the following check boxes (all of them are selected by default):
    - **Coincident entities (blue)**—Show edges and points (in a pure blue color) that lie in the work plane.
    - **Intersection (cyan)**—Show the intersection of 3D geometry and the work plane (in cyan).
    - **Projection (light blue)**—Show the projection of all edges and points onto the work plane (in light blue).

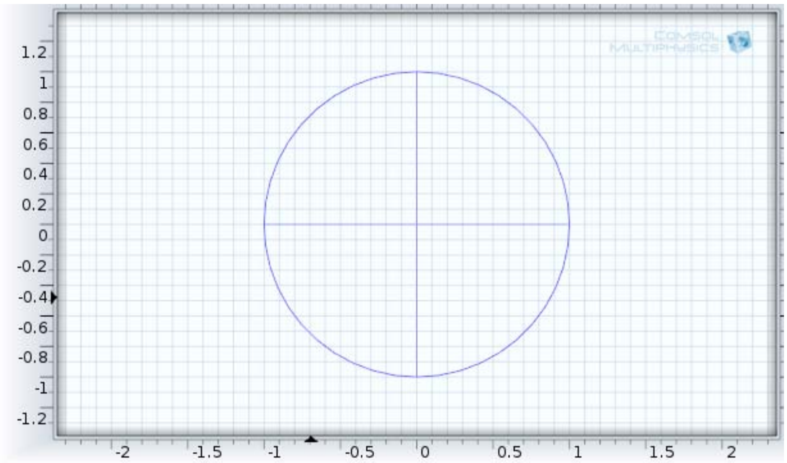


Figure 7-3: The Graphics window displaying the Work Plane Geometry without the Draw on work plane in 3D check box selected.

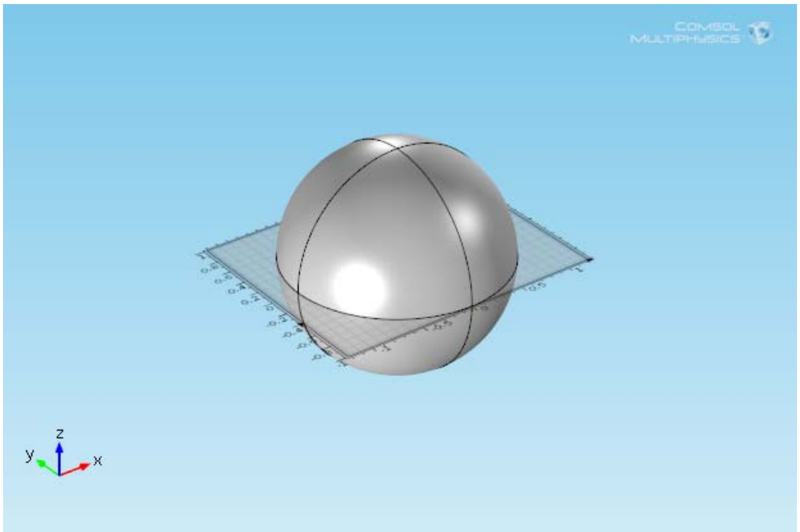


Figure 7-4: The Graphics window displaying the Work Plane Geometry with the Draw on work plane in 3D check box selected.

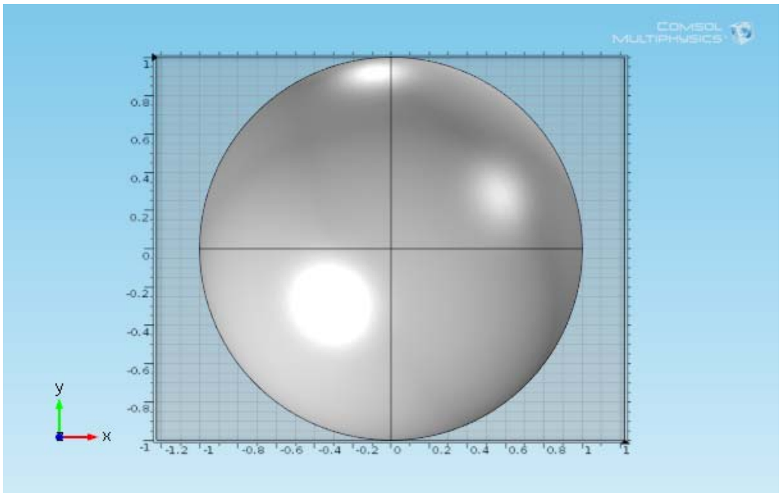


Figure 7-5: Click the Align with Work Plane button to display the geometry from the top down.

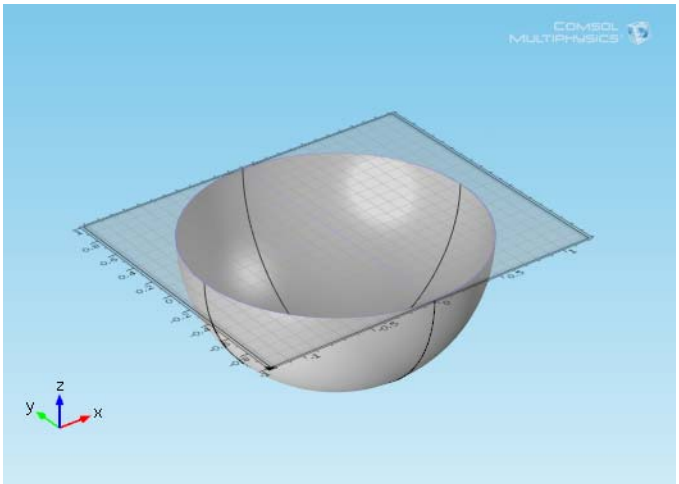


Figure 7-6: Click to turn on the Work Plane Clipping button and cut away all geometries above the work plane.

Virtual Geometry and Mesh Control Operations

Forming Composite Edges and Faces by Ignoring Vertices and Edges and Merging Vertices by Collapsing Edges

VIRTUAL GEOMETRY OPERATIONS

For a 2D or 3D geometry you can add *virtual geometry operations* after the **Form Union/Assembly** node to, for example, remove small details from the geometry and to prepare it for efficient meshing and analysis. The geometry sequence before the **Form Union/Assembly** node defines the “real” (original) geometry. The geometry resulting from a virtual operation is referred to as a *virtual geometry*.

These virtual operations are available from the **Geometry** toolbar, **Virtual Operations** menu ():

TABLE 7-8: VIRTUAL OPERATIONS FOR 2D AND 3D MODELS

ICON	NAME	DESCRIPTION
	Ignore Edges	Virtually remove isolated edges or edges adjacent to precisely two faces or between two domains
	Ignore Faces	Virtually remove isolated faces or faces between two domains (3D only)
	Ignore Vertices	Virtually remove isolated vertices or vertices adjacent to precisely two edges
or	Form Composite Domains	Form virtual composite domains from sets of connected domains by ignoring the boundaries between the domains in each set
	Form Composite Edges	Form virtual composite edges from sets of connected edges by ignoring the vertices between the edges in each set
	Form Composite Faces	Form virtual composite faces from sets of connected faces by ignoring the edges between the faces in each set (3D only)
	Collapse Edges	Virtually collapse each edge into a vertex by merging its adjacent vertices
	Collapse Faces	Virtually collapse faces into edges or vertices by merging their adjacent opposite edges or collapsing all adjacent edges
	Merge Edges	Virtually merge opposite edges adjacent to a face
	Merge Vertices	Virtually merge one adjacent vertex of an edge with the other adjacent vertex

MESH CONTROL OPERATIONS

The following mesh control operations are available are available from the **Geometry>Virtual Operations** menu ( ) and can be used to include ignored geometric entities for mesh control purposes to, for example, make it possible to use a mapped mesh:

TABLE 7-9: MESH CONTROL OPERATIONS

ICON	NAME	DESCRIPTION
	Mesh Control Domains	To select domains for mesh control only.
	Mesh Control Edges	To use isolated edges, or edges adjacent to precisely two domains (in 2D) or two faces (in 3D), only for mesh control.
	Mesh Control Faces	To use isolated faces, or faces between two 3D domains, only for mesh control.
	Mesh Control Vertices	To use isolated vertices, or vertices adjacent to precisely two edges, only for mesh control.

Collapse Edges

To virtually collapse edges into vertices, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Collapse Edges** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to collapse in the Graphics window. They then appear in the **Edges to collapse** list. If the geometry sequence includes user-defined selections above the **Collapse Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to collapse**.

Click the **Active** button to toggle between turning ON and OFF the **Edges to collapse** selections.

Select the **Ignore merged vertices** check box to ignore the resulting merged vertices (if possible).

The operation collapses an edge by removing it, merging its adjacent vertices to the vertex with lowest index, and reconnecting the adjacent edges to the *merged vertex*.

The operation collapses the edges in the selection where no other edge shares the same adjacent vertices (unless this is in the selection).

Collapse Faces

To virtually collapse faces into edges or vertices, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Collapse Faces** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to collapse in the **Graphics** window. They then appear in the **Faces to collapse** list. If the geometry sequence includes user-defined selections above the **Collapse Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to collapse**.

Click the **Active** button to toggle between turning ON and OFF the **Faces to collapse** selections.

The operation collapses a face by removing it, merging its adjacent opposite edges or collapsing all adjacent edges, and reconnecting the adjacent faces to the resulting merged edges.

Select the **Ignore merged entities** check box to ignore the resulting merged edges or vertices (if possible).

Form Composite Domains

To form virtual composite domains, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Form Composite Domains** (or). Then enter the properties of the operation using the following sections:

INPUT

Select the domains that you want to composite in the **Graphics** window. They then appear in the **Domains to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Domains** node, choose **Manual** to select domains, or choose one of the selection nodes from the list next to **Domains to composite**.

Click the **Active** button to toggle between turning ON and OFF the **Domains to composite** selections.

Use the **Ignore adjacent vertices** (2D) or **Ignore adjacent edges and vertices** (3D) check box to specify if the operation also removes the ignorable vertices (and edges in 3D) on the boundary of each resulting composite domain.

The operation forms a composite domain for each connected domain component of the selected domains by ignoring the boundaries between the domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected domains are composed in the geometry but are available individually when you build the mesh. This gives you more control of the meshing. A well partitioned geometry is more efficient to mesh and can, for example, make it possible to create a high quality hexahedral mesh through the sweep operations.

Form Composite Edges

To form virtual composite edges, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Form Composite Edges** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to composite in the **Graphics** window. They then appear in the **Edges to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to composite**.

Click the **Active** button to toggle between turning ON and OFF the **Edges to composite** selections.

The operation forms a composite edge for each connected edge component (of manifold type) of the selected edges by ignoring the vertices between the edges. However, it does not ignore vertices when that would introduce closed or periodic composite edges.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected edges are composed in the geometry but are available individually when you build the mesh.

Form Composite Faces

To form virtual composite faces, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Form Composite Faces** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to composite in the **Graphics** window. They then appear in the **Faces to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to composite**.

Click the **Active** button to toggle between turning ON and OFF the **Faces to composite** selections.

Use the **Ignore adjacent vertices** check box to specify if the operation also removes the ignorable vertices on the boundary of each resulting composite face.

The operation forms a composite face for each connected face component (of manifold type) of the selected faces by ignoring the edges between the faces.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected faces are composed in the geometry but are available individually when you build the mesh.

Ignore Edges

To remove isolated edges or edges adjacent to precisely two domains or two faces from the geometry, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Ignore Edges** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to ignore in the **Graphics** window. These then appear in the **Edges to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to ignore**.

Click the **Active** button to toggle between turning ON and OFF the **Edges to ignore** selections.

Use the **Ignore adjacent vertices** check box to specify if the operation also removes the ignorable start and end vertices of the edges.

The operation removes the selected edges that are isolated or that are adjacent to precisely two faces or are between two domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected edges disappear from the geometry but become available when you build the mesh. You can, for example, use *mesh control edges* to control the element size inside a domain or to partition the geometry to use a mapped mesh. See also Mesh Control Edges.

Ignore Faces

This operation is available for 3D models only.

To remove isolated faces or faces between two domains from the geometry, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Ignore Faces** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to ignore in the **Graphics** window. They then appear in the **Faces to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to ignore**.

Click the **Active** button to toggle between turning ON and OFF the **Faces to ignore** selections.

Use the **Ignore adjacent vertices and edges** check box to specify if the operation also removes the ignorable vertices and edges adjacent to the faces.

The operation removes the selected faces that are isolated or that are between two domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected faces disappear from the geometry but become available when you build the mesh. You can, for example, use a *mesh control face* to partition the geometry to make it possible to sweep a hexahedral mesh. See also Mesh Control Faces.

Ignore Vertices

To remove isolated vertices or vertices adjacent to two edges only from the geometry, on the **Geometry** toolbar, from the **Virtual Operations** menu (), select **Ignore Vertices** (). Then enter the properties of the operation using the following sections:

INPUT

Select the vertices (points) that you want to ignore in the **Graphics** window. These then appear in the **Vertices to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Vertices** node, choose **Manual** to select vertices, or choose one of the selection nodes from the list next to **Vertices to ignore**.

Click the **Active** button to toggle between turning ON and OFF the **Vertices to ignore** selections.

The operation removes the selected vertices that are isolated or that are adjacent to precisely two edges. However, it does not remove vertices if that would introduce closed or periodic *composite edges*.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected vertices disappear from the geometry but become available when you build the mesh. You can, for example, use a *mesh control vertex* to control the element size inside a domain. See also Mesh Control Vertices.

Merge Edges

To virtually merge opposite edges adjacent to a face, on the **Geometry** toolbar, from the **Virtual Operations** menu (), select **Merge Edges** (). Then enter the properties of the operation using the following sections:

EDGES TO KEEP

Select the edges that you want to keep in the **Graphics** window. They then appear in the **Edges to keep** list.

Click the **Active** button to toggle between turning ON and OFF the **Edges to keep** selections.

EDGES TO REMOVE

Select the edges that you want to remove in the **Graphics** window. They then appear in the **Edges to remove** list. Click the **Active** button to toggle between turning ON and OFF the **Edges to remove** selections.

The operation merges the opposite edges by collapsing the face between the edges and reconnecting the faces adjacent to the removed edges to the resulting merged edges.

Merge Vertices

To virtually merge one vertex adjacent to an edge with the other adjacent vertex, on the **Geometry** toolbar, from the **Virtual Operations** menu (), select **Merge Vertices** (). Then enter the properties of the operation using the following sections:

VERTEX TO KEEP

Select the vertex that you want to keep in the **Graphics** window. It then appears in the **Vertex to keep** list.

Click the **Active** button to toggle between turning ON and OFF the **Vertex to keep** selections.

VERTEX TO REMOVE

Select the vertex that you want to remove in the **Graphics** window. It then appears in the **Vertex to remove** list. Click the **Active** button to toggle between turning ON and OFF the **Vertex to remove** selections.

The operation merges the two vertices by collapsing the edge between the vertices and reconnecting the edges adjacent to the removed vertex to the resulting merged vertex.

Mesh Control Domains

To use a domain only for mesh control, on the **Geometry** toolbar, from the **Virtual Operations** menu (), select **Mesh Control Domains** (). Then enter the properties of the operation using the following sections:

INPUT

Select the domains that you want to use for mesh control in the Graphics window. They then appear in the **Domains to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Domains** node, choose **Manual** to select domains, or choose one of the selection nodes from the list next to **Domains to include**.

Click the **Active** button to toggle between turning ON and OFF the **Domains to include** selections.

The operation removes the selected domains by composing them with adjacent domains. The faces (3D only) and edges in between are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control domain* to partition the geometry to make it possible to sweep a hexahedral mesh.

Mesh Control Edges

To use isolated edges, or edges adjacent to precisely two domains (in 2D) or two faces (in 3D), only for mesh control, on the **Geometry** toolbar, from the **Virtual Operations** menu (), select **Mesh Control Edges** (). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to use for mesh control in the **Graphics** window. They then appear in the **Edges to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to include**.

Click the **Active** button to toggle between turning ON and OFF the **Edges to include** selections.

Use the **Include adjacent vertices** check box to specify if the operation also include the ignorable start and end vertices of the edge.

The operation removes the selected edges that are isolated or that are adjacent to precisely two faces (in 3D) or two domains (in 2D). The edges are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control edge* to control the element size inside a domain or to partition the geometry to make use of a mapped mesh.

Ignore Edges

Mesh Control Faces

To use isolated faces, or faces between two 3D domains, only for mesh control, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Mesh Control Faces** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to use for mesh control in the **Graphics** window. They then appear in the **Face to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Feces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to include**.

Click the **Active** button to toggle between turning ON and OFF the **Faces to include** selections.

Use the **Include adjacent vertices and edges** check box to specify if the operation also includes the ignorable vertices and edges adjacent to of the faces.

The operation removes the selected faces that are isolated or that are between two domains. The faces are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control face* to partition the geometry to make it possible to sweep a hexahedral mesh.

Ignore Faces

Mesh Control Vertices

To use isolated vertices, or vertices adjacent to precisely two edges, only for mesh control, on the **Geometry** toolbar, from the **Virtual Operations** menu ( ), select **Mesh Control Vertices** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the vertices (points) that you want to use for mesh control in the **Graphics** window. They then appear in the **Vertices to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Vertices** node, choose **Manual** to select vertices, or choose one of the selection nodes from the list next to **Vertices to include**.

Click the **Active** button to toggle between turning ON and OFF active **Vertices to include** selections.

The operation removes the selected vertices that are isolated or that are adjacent to precisely two edges. However, it does not remove vertices if that would introduce closed or periodic *composite edges*. The selected vertices are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control vertex* to control the element size inside a domain.

Ignore Vertices

Geometry Modeling Examples

Creating a 2D Geometry Model

This section describes how to build a 2D cross section of a heat sink and introduces 2D geometry operations in COMSOL Multiphysics. At this time, you do not model the physics that describe the operation of the heat sink.

Assume that you want to estimate the maximum amount of heat dissipated by a heat sink placed around a resistor for high-power applications. The heat sink consists of an extruded aluminum profile as in Figure 7-7. If the effects at the ends of the elongated heat sink are neglected, the model can be simplified and a decent estimate obtained of the heat dissipated by creating a 2D cross section.

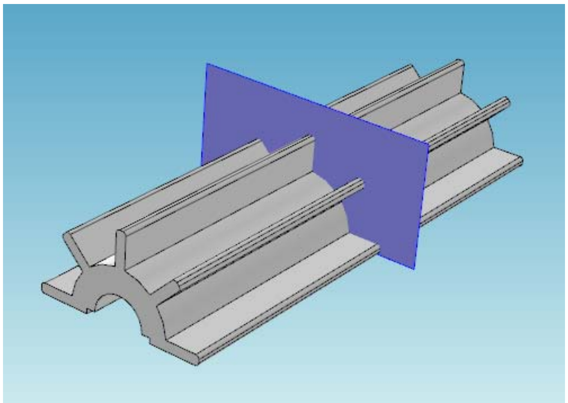


Figure 7-7: Example of a 3D heat sink model with cross section.

CREATING A BASIC 2D GEOMETRY MODEL

1 Double-click the COMSOL Multiphysics icon to launch COMSOL.

2 Add a **2D** Component, either when Creating a New Model or from The Component Node.

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

The following steps explain how to create two circles to form the core of the heat sink in Figure 7-7. To investigate different dimensions of the heat sink, parameterize the geometry. Start by defining the radius of the outer arc of the heat sink, the radius of the inner arc, and the thickness and the length of the heat sink flanges.

See Toolbars and Keyboard Shortcuts for links and information about all the available toolbars. Also see The COMSOL Desktop Menus and Toolbars.

It is also useful to review Working with Geometric Entities and Named Selections before continuing with these instructions.

1 On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Parameters** (Pi).

2 In the **Parameters** table, enter, or copy and paste the values in the table below. The **Value** column automatically displays the **Expression** value.

NAME	EXPRESSION	DESCRIPTION
`R1`	`5[mm]`	`Radius circle 1`
`R2`	`2.5[mm]`	`Radius circle 2`

NAME	EXPRESSION	DESCRIPTION
d	1[mm]	Height
L	5[mm]	Width

ADDING TWO CIRCLES WITH PREDEFINED PARAMETERS

1 On the Geometry Toolbar add a **Circle** (○).

2 On the **Circle** settings window, under **Size and Shape**, enter `R1` in the **Radius** field.

3 Click the **Build Selected** button (▦).

A circle (`c1`) with radius `R1` displays in the **Graphics** window.

4 Add another circle. Right-click **Geometry 1** and select **Circle** (○).

5 On the **Circle** settings window, under **Size and Shape**, enter `R2` in the **Radius** field.

6 Click the **Build Selected** button (▦).

A circle with radius `R2` displays in the **Graphics** window. Click the **Zoom Extents** button (⊕) to see both circles.

SUBTRACTING THE SMALLER CIRCLE

1 On the **Geometry** toolbar, click **Difference** (⧉).

2 On the settings window, under **Difference**, the **Active** button is **On** (ON) by default. It activates the **Objects to add** list for choosing objects.

3 In the **Graphics** window, select the object **c1** (the larger circle) by hovering the mouse the object and left-clicking it.

`c1` is added to the **Objects to add** list.

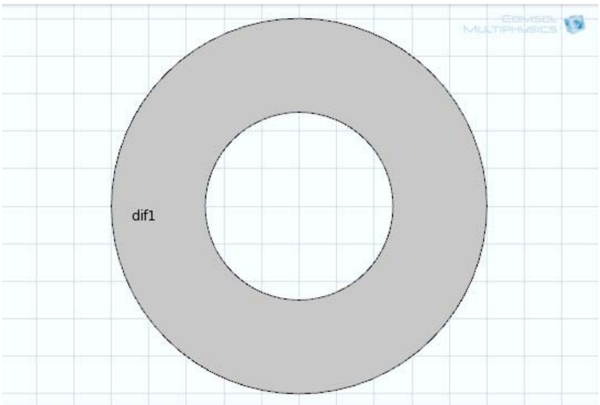
4 Under **Objects to subtract** click the **Active** button to toggle ON and activate this section.

5 Select the object `c2` (the smaller circle) by hovering the mouse the object and left-clicking it.

`c2` is added to the **Objects to subtract** list.

6 Click the **Build Selected** button (▦).

The object **dif1** is created by subtracting the smaller circle from the larger circle.



INTERSECTING WITH RECTANGLE

1 On the **Geometry** toolbar, add a **Rectangle** (▭).

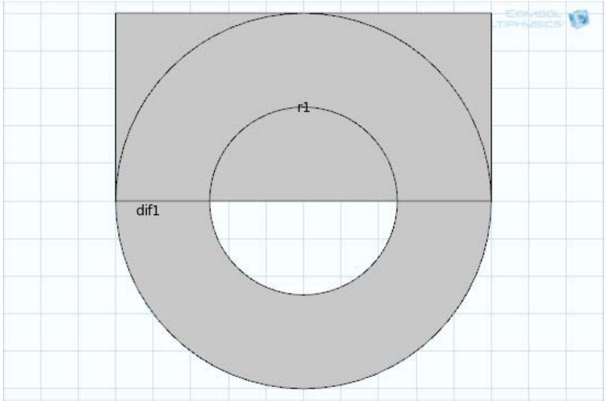
2 On the settings window, under **Size**:

a In the **Width** field enter `2*R1`, and in the **Height** field, enter `R1`.

b Under **Position**, enter `-R1` in the **x** field.

3 Click the **Build Selected** button (▦).

The interaction operation creates the object `r1` (not related to the circle radius), which coincides with the intersecting area of the two input objects.

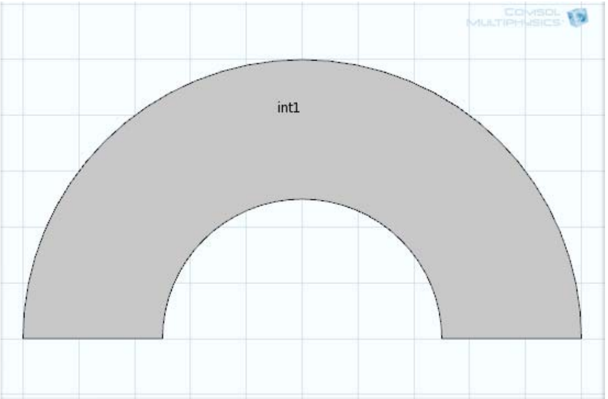


4 On the **Geometry** toolbar, click **Intersection** (▣).

5 In the **Graphics** window, click to select each object, `dif1` (the combined circle) and `r1` (the rectangle).

After each click, the object is added to the **Input Objects** list.

6 Click the **Build Selected** button (▦) to create the object `int1`.



ADDING A RECTANGLE TO CREATE A FLANGE

1 On the **Geometry** toolbar, add a **Rectangle** (□).

2 On the **Rectangle** settings window, under **Size**:

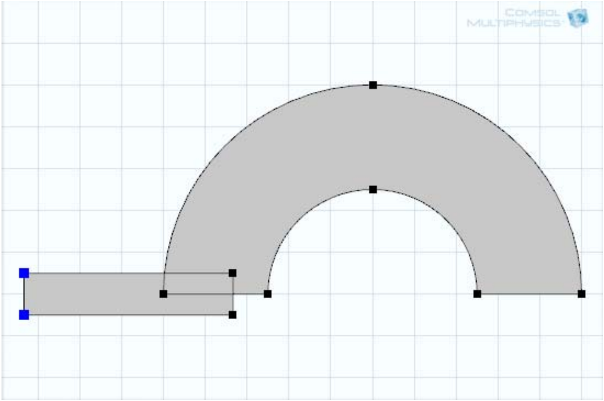
a In the **Width** field, enter `L`.

b In the **Height** field, enter `d`.

3 Under **Position**, in the **x** field enter `-(2/3*R1+L)`, and in the **y** field enter `-d/2`.

4 Click the **Build Selected** button (▦).

The object `r2` (not related to the circle radius) is created. Next, round the sharp edges of the flange by using fillets. On the **Graphics** window toolbar, click the **Zoom Extents** button (⊞).



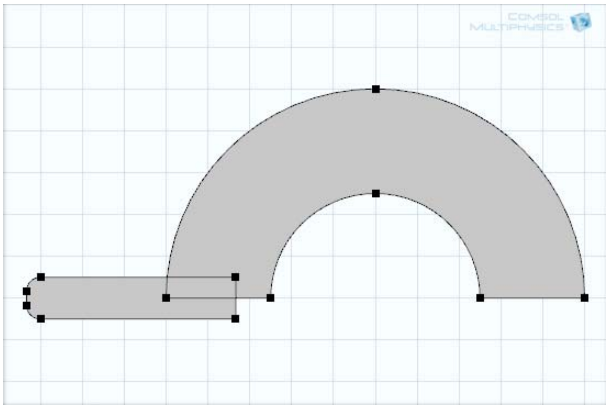
ADDING A FILLET TO ROUND THE FLANGE EDGES

1 On the **Geometry** toolbar click **Fillet** (◜).

2 On object `r2` (the small rectangle) click each vertex (**1** and **4**, located in the left-hand corners, as highlighted in the previous figure) to add these to the selection lists.

3 On the **Fillet** settings window, under **Radius**, enter `d/3` in the **Radius** field.

4 Click the **Build Selected** button (▦) to create object `fil1`.



ADDING ROTATE OPERATIONS TO CREATE FIVE FLANGES

Rotate the flange 45 degrees and keep the original input object to create five flanges on top of the heat sink.

Adding Rotate 1 to Create Object Rot1

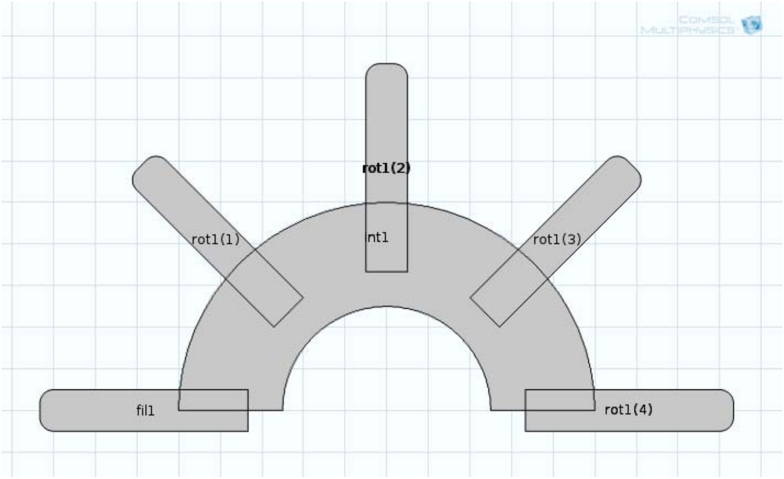
1 On the **Geometry** toolbar, click **Rotate** (↻).

2 In the **Graphics** window, click to select object `fil1` (the filleted rectangle). It is added to the **Input Objects** list.

3 On the **Rotate** settings window, under **Input**, select the **Keep input objects** check box.

4 Under **Rotation Angle**, enter `-45 -90 -135 -180` in the **Rotation** field.

5 Click **Build Selected** (▦) to create the object `rot1`. Click **Zoom Extents**(⊞).



REMOVING INTERIOR BOUNDARIES IN UNION OPERATIONS

1 On the **Geometry** toolbar, click **Union** ( ).

2 Select the objects `int1`, `fil1`, `rot1(1)`, `rot1(2)`, `rot1(3)`, and `rot1(4)`.

3 Under **Union**, click to clear the **Keep interior boundaries** check box to remove the interior boundaries in the union operation. This is good practice if these boundaries do not define separate parts with different materials, for example.

4 Click the **Build All Objects** button ( ). Click the **Zoom Extents** button ( ). The final geometry is shown in Figure 7-8.

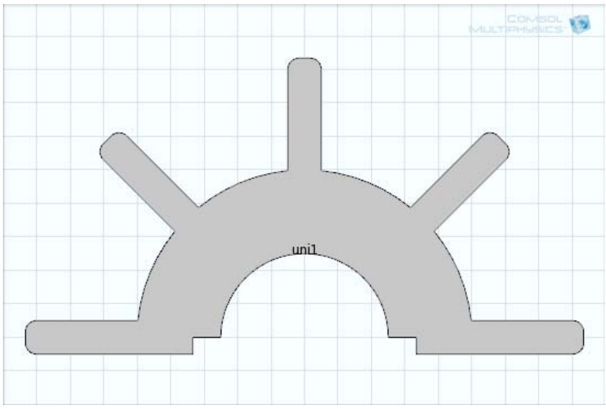
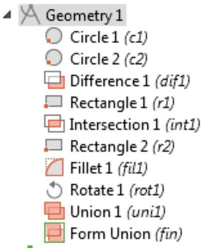


Figure 7-8: Final 2D object created in the Model Builder.

VIEWING THE GEOMETRY SEQUENCE

Figure 7-9 shows the geometry sequence used to create Figure 7-8. All primitive objects and the fillet operation are parameterized through the radius of the inner and outer heat sink arcs, the length and thickness of the flanges, and the radius of the fillets. You can change the parameter values in the **Parameters** table or for any object to create alternative heat sink geometries. The sequence still remains, and when you click the **Build All** button ( ) a new geometry is created.



Geometry 1
- Circle 1 *(c1)*
- Circle 2 *(c2)*
- Difference 1 *(dif1)*
- Rectangle 1 *(r1)*
- Intersection 1 *(int1)*
- Rectangle 2 *(r2)*
- Fillet 1 *(fil1)*
- Rotate 1 *(rot1)*
- Union 1 *(uni1)*
- Form Union *(fin)*

Figure 7-9: An example of a 2D geometry sequence.

RE-RUNNING THE GEOMETRY SEQUENCE WITH DIFFERENT PARAMETERS

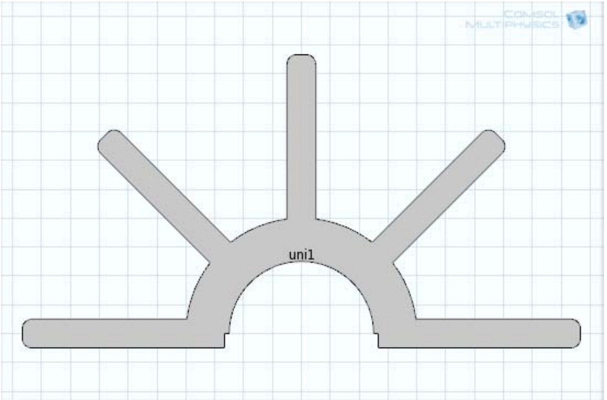
1 On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Parameters** (Pi).

2 On the settings window, under **Parameters**, enter the following settings in the table. Replace the previous data:.

NAME	EXPRESSION	VALUE	DESCRIPTION
R1	`4[mm]`	0.0040 m	Radius Circle 1
R2	`2.5[mm]`	0.0025 m	Radius Circle 2
d	`1[mm]`	0.0010 m	Height
L	`7[mm]`	0.0070 m	Width

3 In the **Model Builder**, click **Geometry 1**.

4 Click the **Build All** button (🟩). Click the **Zoom Extents** button ( ) to view the geometry as defined by the new parameters.



Creating a 3D Geometry Model

Figure 7-10 shows the geometry of a heat sink used for cooling microprocessors. The following sections describe the steps to create this geometry and introduces 3D drawing tools and techniques.

> See Toolbars and Keyboard Shortcuts for links and information about all the available toolbars. Also see The COMSOL Desktop Menus and Toolbars.
>
> It is also useful to review Working with Geometric Entities and Named Selections before continuing with these instructions.

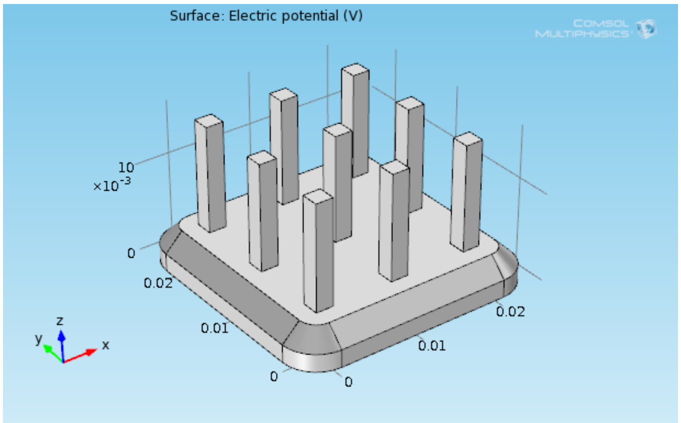


Figure 7-10: Example of a 3D heat sink model.

CREATING A BASIC 3D GEOMETRY COMPONENT

1 Double-click the COMSOL Multiphysics icon to launch COMSOL.

2 Add a **3D** Component, either when Creating a New Model or from The Component Node.

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

1 On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Parameters** (Pi).

2 On the settings window, in the **Parameters** table, enter these settings:

NAME	EXPRESSION	VALUE	DESCRIPTION
L1	1.5e-2	0.015	Pillar thickness (in the heat sink)
L2	3e-3	0.0030	Pillar length (in the heat sink)

USING WORK PLANES TO CREATE A BÉZIER POLYGON

Use work planes to create 2D geometries that you can extrude or revolve to create 3D objects.

Work Plane and Using Work Planes

Creating a Bézier Polygon

1 On the **Geometry** toolbar, click **Work Plane** ( ).

2 On the **Work Plane** settings window, under **Plane Definition**, select **xz-plane** from the **Plane** list.

3 Under the **Work Plane 1** node, right-click **Plane Geometry** and add a **Bézier Polygon** ( ).

4 On the **Bézier Polygon** settings window, under **Polygon Segments**, click **Add Linear**.

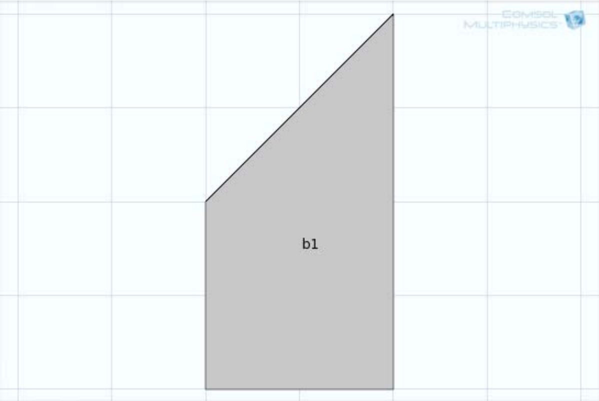
Segment 1 (linear) displays in the **Added segments** list.

5 Under **Control points**: In row **1**, enter -2e-3 in the **xw** field, and in row **2**, enter –4e-3 in the **xw** field.

6 Click **Add Linear** to add **Segment 2 (linear)** to the **Added segments** list. Some of the **Control points** are automatically filled in with values; the control points from the previous line are already filled in as the starting points for the next line.

7 Under **Control points**, in row **2**, enter 2e-3 in the **yw** field.

8. Click **Add Linear** to add **Segment 3 (linear)** to the **Added segments** list. In row **2**, enter `-2e-3` in the **xw** field, and in row **2**, enter `4e-3` in the **yw** field.
9. Click **Add Linear** to add **Segment 4 (linear)** to the **Added segments** list.
10. Under **Control points**, in row **2**, enter `0` in the **yw** field.
11. Click **Close Curve** then click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).



REVOLVING A 2D OBJECT TO CREATE A 3D OBJECT

1. On the **Geometry** toolbar, click **Revolve** ( ).

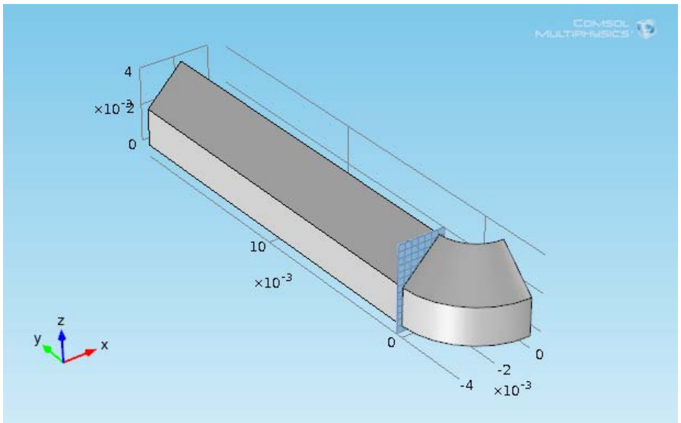
   The **Revolve** settings window opens and the 2D Bézier Polygon displays in the **Graphics** window.
2. On the **Revolve** settings window, under **Revolution Angles**, enter `90` in the **End angle** field.

> The **Revolution Axis** corresponds to the position of the *y*-axis in the work plane's 2D coordinate system.

3. Under **General**, click to clear the **Unite with input objects** check box. **Work Plane 1** is required for the next steps.
4. Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ) to view the object **rev1**.

ADDING AN EXTRUSION AND UNION

1. On the **Geometry** toolbar, click **Extrude** ( ).
2. On the settings window, under **Distances from Plane**, enter `-2e-2` in the **Distances** row.
3. Click the **Build Selected** button ( ) and the **Zoom Extents** button ( )to view the object **ext1**.



4 On the **Geometry** toolbar, click **Union** ().

5 In the **Graphics** window, click to select the objects **rev1** and **ext1** and add them to the **Input objects** section.

6 On the **Union** page, under **Union**, click to clear the **Keep interior boundaries** check box to remove the interior boundary between the corner section and the edge section.

7 Click the **Build Selected** button ( ). Objects **rev1** and **ext1** are combined to create object **uni1**.

ADDING A ROTATION TO THE 3D OBJECT

1 In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ( ).

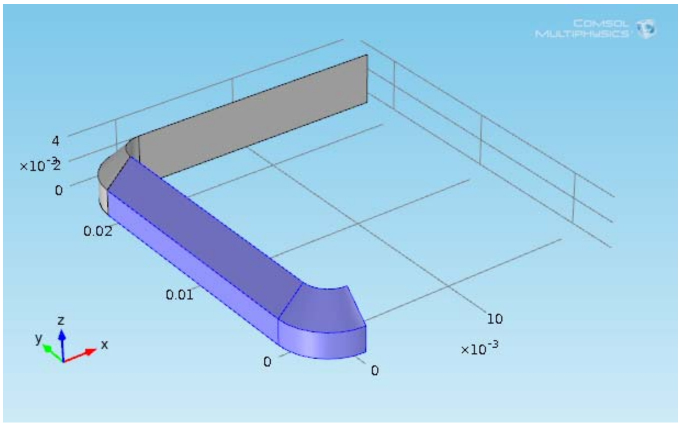
2 Select the object **uni1** and add it to the **Input objects** section under **Input**.

3 Select the **Keep input objects** check box to leave the input object intact as a rotation of the object is created.

4 Under **Rotation Angle**, enter `-90` in the **Rotation** field.

5 Under **Point on Axis of Rotation**: In the **x** field, enter `1e-2`, and in the **y** field, enter `1e-2`.

6 Click the **Build Selected** button ( ) and the **Zoom Extents** button () to view the object **rot1**.



CREATING UNION 2

1 On the **Geometry** toolbar, click **Union** ().

2 In the **Graphics** window, click to select the objects **uni1** and **rot1** and add them to the **Input objects** section under **Union**.

3 Click to clear the **Keep interior boundaries** check box.

4 Click the **Build Selected** button ( ) to create object **uni2**.

ADDING A SECOND ROTATION

1 On the **Geometry** toolbar, click **Rotate** ( ).

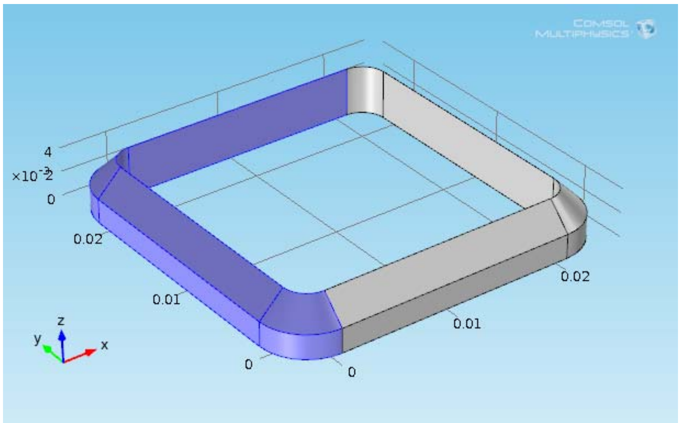
2 In the **Graphics** window, click to select the object **uni2** and add it to the **Input objects** section under **Input**.

3 On the **Rotate** settings window, click to select the **Keep input objects** check box.

4 Under **Rotation Angle**, enter `-180` in the **Rotation** field.

5 Under **Point on Axis of Rotation**: In the **x** field, enter `1e-2`, and in the **y** field, enter `1e-2`.

6 Click the **Build Selected** button ( ).



CREATING UNION 3

1 On the **Geometry** toolbar, click **Union** ( ).

2 In the **Graphics** window, click to select the objects **uni2** and **rot2** and add them to the **Input objects** section under **Union**.

3 Click to clear the **Keep interior boundaries** check box.

4 Click the **Build Selected** button ( ) to create object **uni3**.

CREATING WORK PLANE 2 AND ADDING A SQUARE

1 On the **Geometry** toolbar, click **Work Plane** ( ). A **Work Plane 2** node is added to the **Model Builder**.

2 Click the **Build Selected** button ( ).

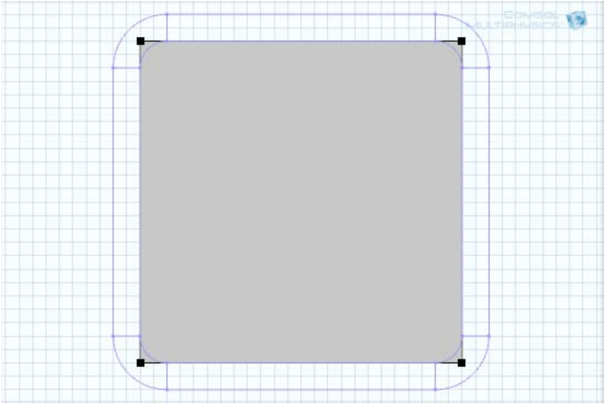
3 On the **Work Plane 2** settings window, in the upper left corner, click the **Show Work Plane** button ( ). Use the projection of the 3D geometry on the *xy*-plane as a guide for creating the middle section of the heat sink base.

4 In the **Model Builder**, under **Work Plane 2**, right-click **Plane Geometry** and select **Square** ( ).

5 On the **Square** settings window, under **Size**, enter `2.4e-2` in the **Side length** field.

6 Under **Position**: Select **Center** from the **Base** list. Then in the **xw** field, enter `1e-2`, and in the **yw** field, enter `1e-2`.

7 Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).



TRIMMING THE SQUARE TO FIT USING THE FILLET OPERATION

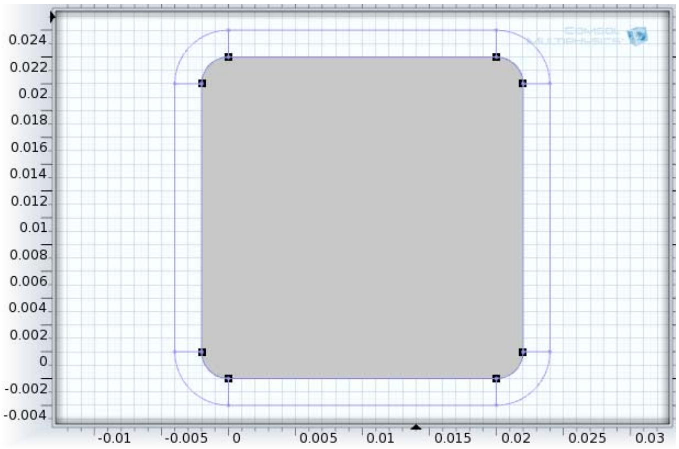
1 Under **Work Plane 2**, click the **Plane Geometry** node.

2 On the Work Plane Modal Toolbar, click **Fillet** ( ).

3 In the **Graphics** window, click to add points 1, 2, 3, and 4 on the object **sq1** to the **Vertices to fillet** section under **Points**.

4 Under **Radius**, enter `2e-3` in the **Radius** field.

5 Click the **Build Selected** button.( ).



ADDING EXTRUDE 2 AND COMBINING OBJECTS TO COMPLETE THE BASE

1 On the **Geometry** toolbar, click **Extrude** ( ).

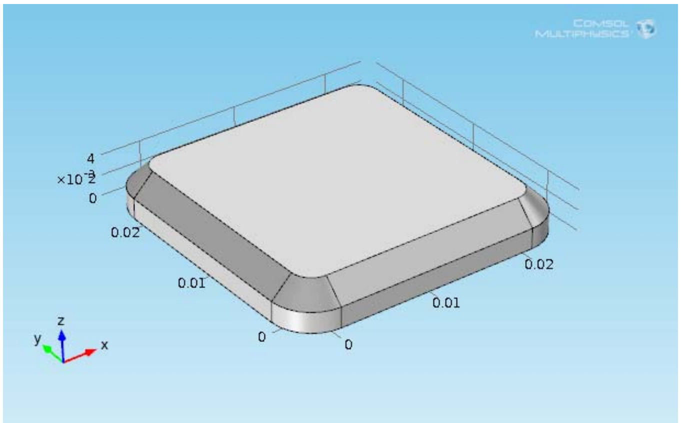
2 On the **Extrude** settings window, under **Distances from Plane**, enter `4e-3` in the **Distances** row.

3 Click the **Build Selected** button ( ).

4 On the **Geometry** toolbar, click **Union** ( ).

5 On the **Graphics** window, click to select the objects **uni3** and **ext2** to add to the **Input objects** section.

6 Click the **Build Selected** button (▦) to create object **uni4**. This completes the base of the heat sink.



DRAWING THE UPPER PART OF THE HEAT SINK

Creating a Work Plane and a Square

1 On the **Geometry** toolbar, click **Work Plane** ( ).

A **Work Plane 3** node is added to the **Model Builder**.

2 On the settings window, under **Work Plane**, enter `4e-3` in the **z-coordinate** field.

3 Click the **Work Plane** node under **Work Plane 3**. In the settings window, the check boxes **Coincident entities**, **Intersection**, and **Projection** are selected by default. This visualizes the projected edges of the heat sink's base in the work plane.

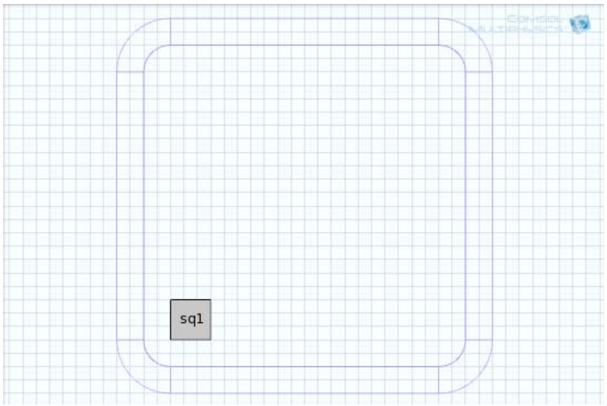
4 Click the **Build Selected** button (▦).

5 On the **Work Plane** settings window, click the **Show Work Plane** button ( ).

6 In the **Model Builder**, under **Work Plane 3**, right-click **Plane Geometry** and select **Square** (□).

7 Under **Size**, enter `L2` in the **Side length** field.

8 Click the **Build Selected** button (▦) to create square **sq1** with side length L2.



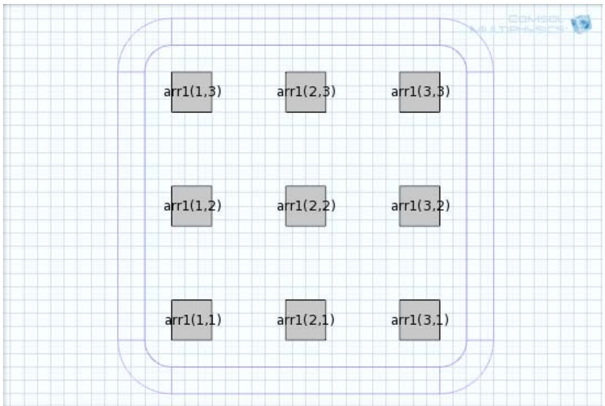
ADDING AN ARRAY OF PILLARS

1 In the **Model Builder**, under **Work Plane 3**, right-click **Plane Geometry** and select **Transforms>Array** ( ).

2 Add the object **sq1** to the **Input objects** section under **Input**.

3 Under **Size**: In the **xw size** field, enter `3`, and in the **yw size** field, enter `3`.

4. Under **Displacement**: In the **xw** field, enter `1e-2-L2/2`, and in the **yw** field, enter `1e-2-L2/2`.
5. Click the **Build Selected** button ( ).



Adding Extrude 3 and Combining Objects (Union)

1. On the **Geometry** toolbar, click **Extrude** ( ).
2. On the **Extrude** settings window, under **Distances from Plane**, enter `L1` in the **Distances** row.
3. Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).
4. Right-click **Geometry 1** again and select **Boolean Operations>Union** ( ).
5. Add all the objects (`uni4` and all the `ext3` just built) to the **Input objects** list under **Union**.
6. Click the **Build All Objects** button ( ) to complete the heat sink geometry.

The upper part of the heat sink is parameterized through the thickness and height of the heat sink pillars. You can edit the parameter values defined previously to change the heat sink geometry.

RE-RUNNING THE GEOMETRY SEQUENCE WITH DIFFERENT PARAMETERS

1. On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Parameters** (Pi).
2. On the settings window, under **Parameters** enter the following settings in the table. Replace the previous data:

NAME	EXPRESSION	VALUE	DESCRIPTION
`L1`	`1.2e-2`	0.012	Pillar thickness (in the heat sink)
`L2`	`2e-3`	0.0020	Pillar length (in the heat sink)

3. In the **Model Builder**, click **Geometry 1**.
4. Click the **Build All** button ( ) and the **Zoom Extents** button ( ) to view the geometry as defined by the new parameters.

Forming Composite Edges and Faces by Ignoring Vertices and Edges

This example of how to use virtual geometry operations shows how to use the Ignore Vertices operation (or the Form Composite Edges operation) to remove a very short edge and how to use the Ignore Edges operation (or the Form Composite Faces operation) to prepare the geometry for swept meshing.

GEOMETRY 1

1. Add a **3D** Component, either when Creating a New Model or from The Component Node.
2. On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Import** ( ).

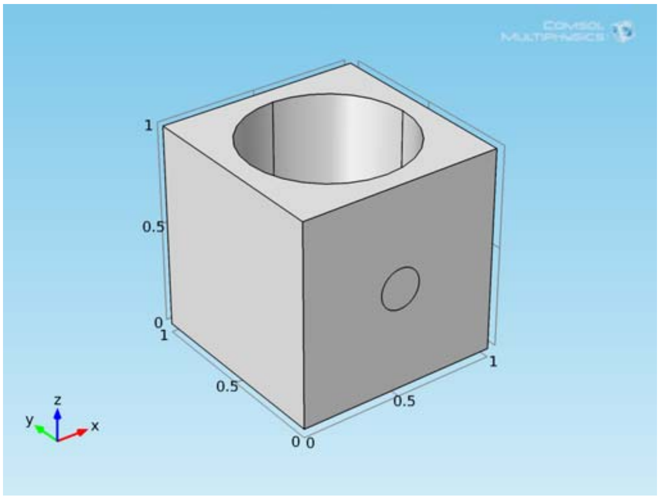
3 On the **Import** settings window, under the **Import** section, click **Browse**.

4 In the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click `virtualgeom_demo_1.mphbin`.

The location of the file varies based on the installation. For example, if the installation is on your hard drive, the file path might be similar to `C:\Program Files\COMSOL44\models`.

5 Click **Import**. On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Build All** (▦).

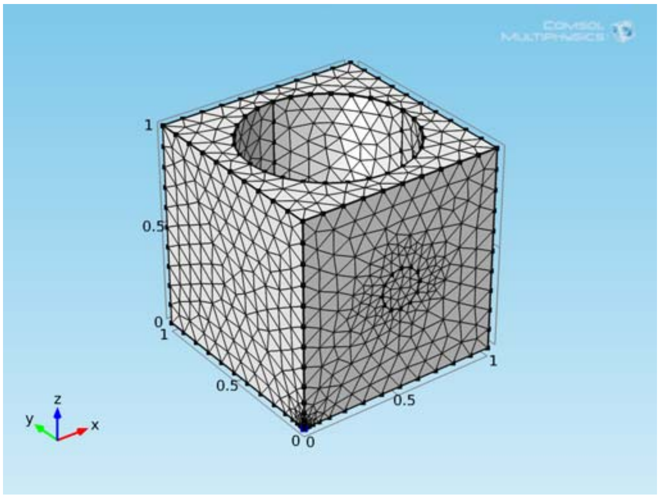
The imported geometry displays in the **Graphics** window.



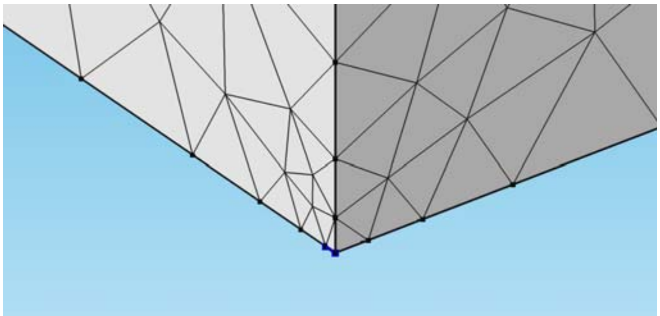
MESH 1

In the **Model Builder**, click the **Mesh** node. On the **Mesh** toolbar, click **Build All** (▦).

The resulting mesh displays in the **Graphics** window.



A **Warning** node (⚠) is added under **Mesh 1** indicating that there is a very short edge in the geometry. Use the **Zoom selected** button (⊕) and the **Zoom out** button (⊖) on the **Graphics** toolbar to locate this edge.



Eliminate the short edge by ignoring the vertex between this edge and its adjacent longer edge.

GEOMETRY 1

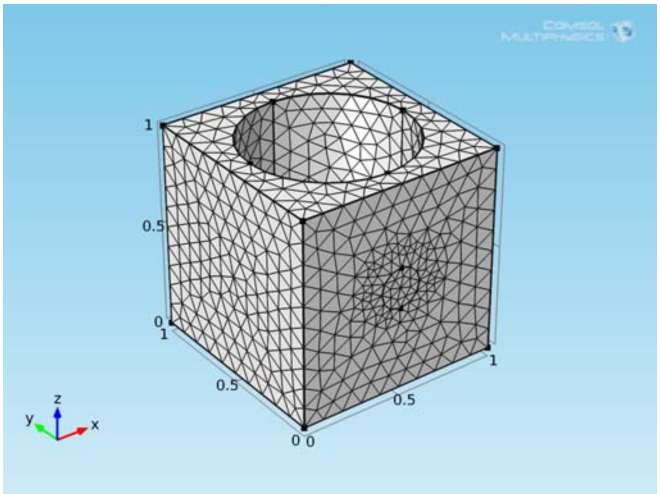
Ignore Vertices 1

1. On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Ignore Vertices** ( ).
2. In the **Graphics** window, click to select Point 3.
3. Click **Build Selected** ( ).

MESH 1

In the **Model Builder**, click **Mesh 1** and on the settings window, click **Build All** ( ).

The mesh displays in the **Graphics** window.



The geometry's domain is well suited for swept meshing.

Swept 1

1. On the **Mesh** toolbar, click **Swept** ( ).
2. On the Swept settings window, click to expand the **Source Faces** section.
3. The **Active** button is ON by default. In the **Graphics** window, click to select Boundary 3. Or click the **Paste** button ( ) and enter 3.
4. Click to expand the **Destination Faces** section.The **Active** button is ON by default. Click to select Boundary 4.

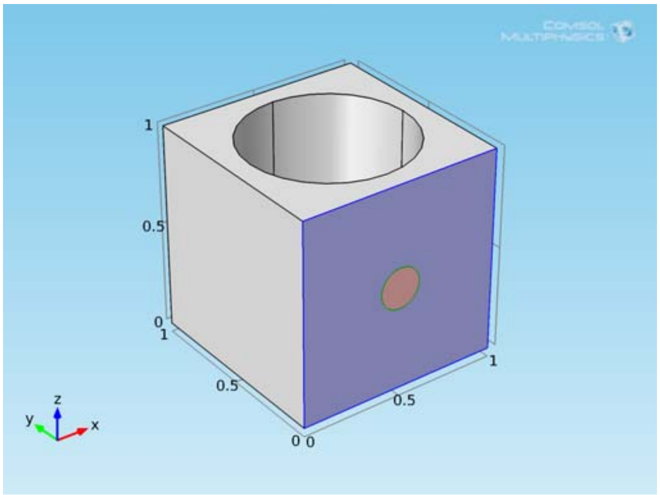
Size

1. In the **Model Builder**, click the **Size** node ( ).

2 In the **Size** settings window, under **Element Size**, choose **Finer** from the **Predefined** list.

3 Click the **Build All** button ( ).

4 Click **OK** to close the **COMSOL Error** window. COMSOL fails to create a swept mesh due to the circular imprint on one of the linking faces of the sweep.



Use the Ignore Edges operation to remove this imprint.

GEOMETRY 1

Ignore Edges 1

1 On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Ignore Edges** ( ).

2 Select Edges 14 and 15.

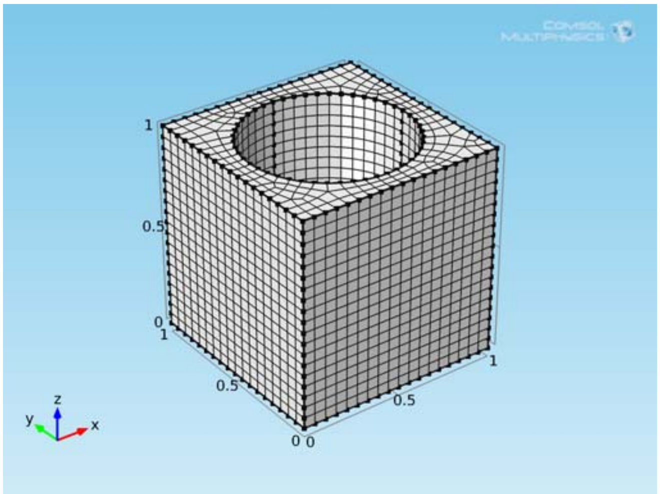
Make sure to select the **Ignore adjacent vertices** check box so that the vertices of the imprint disappear as well.

3 Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Component 1>Mesh 1** and choose **Build All** ( ).

The swept mesh displays in the **Graphics** window.



You can achieve the same virtual geometry using Form Composite Edges and Form Composite Faces operations.

GEOMETRY 1

Disable

1 In the **Model Builder**, right-click **Ignore Edges 1** and choose **Disable** ( ).

2 Right-click **Ignore Vertices 1** and choose **Disable** ( ).

Form Composite Edges 1

1 On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Form Composite Edges** ( ).

2 Select Edges 2 and 6.

3 Click the **Build Selected** button ( ).

Form Composite Faces 1

1 On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Form Composite Faces** ( ).

2 Select Boundaries 2 and 8.

3 Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Component 1>Mesh 1** and choose **Build All** ( ).

Merging Vertices by Collapsing Edges

This example of virtual geometry operations illustrates how you can use the Collapse Edges operation (or the Merge Vertices operation) to prepare the geometry for efficient meshing.

GEOMETRY 1

1 Add a **3D** Component, either when Creating a New Model or from The Component Node.

2 On the **Home** ribbon (Windows) and the **Model Toolbar** (Mac and Linux) click **Import** ( ).

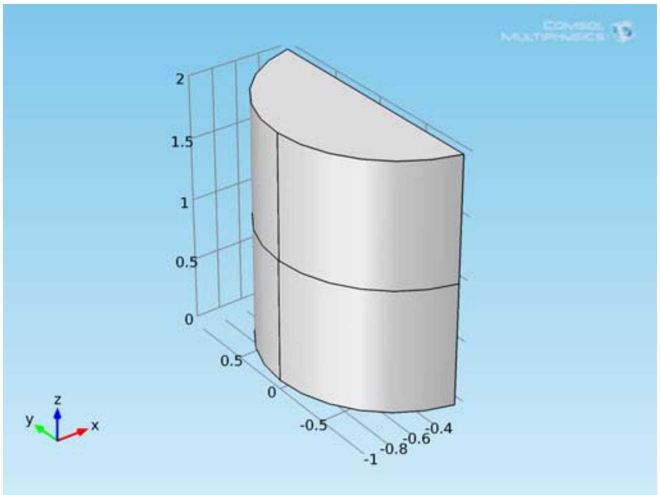
3 On the **Import** settings window, under the **Import** section, click **Browse**.

4 In the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click `virtualgeom_demo_2.mphbin`.

The location of the file varies based on the installation. For example, if the installation is on your hard drive, the file path might be similar to `C:\Program Files\COMSOL44\models`.

5 Click **Import**.

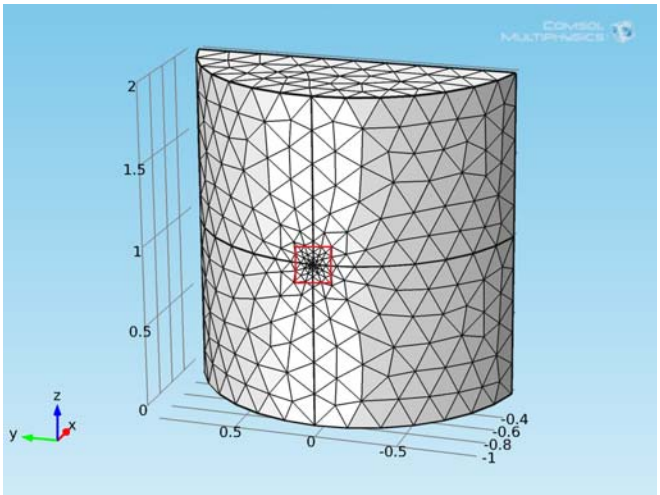
The imported geometry displays in the **Graphics** window.



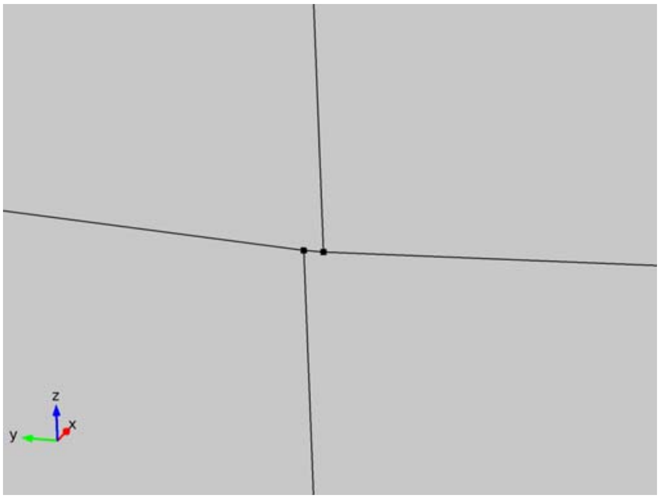
GEOMETRY 1

In the **Model Builder**, click **Mesh 1** and on the settings window, click **Build All** (🟩).

The resulting mesh displays in the **Graphics** window.



The figure shows that the mesh is very fine in the region marked by the red box. If you select **Component 1>Geometry 1** in the **Model Builder** and zoom into this region you can find the reason for the fine mesh. There is a very short edge at the junction of the four curved faces.



Remove this short edge by collapsing it into a vertex.

Collapse Edges 1

1 On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Collapse Edges** ( ).

2 Select Edge 4.

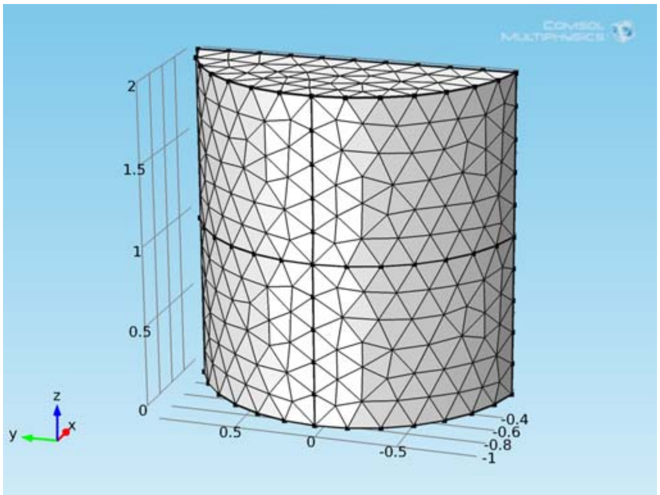
3 Click the **Build Selected** button ( ).

> You can also remove this short edge with an Ignore Edges operation forming a composite face of the two adjacent faces. However, then the small distance between the vertices of the ignored edge remains.

MESH 1

In the **Model Builder**, click **Mesh 1** and on the settings window, click **Build All** ( ).

The mesh displays in the **Graphics** window.



You can achieve the same virtual geometry using a Merge Vertices operation.

GEOMETRY 1

Right-click **Collapse Edges 1** and choose **Disable** (⊘).

Merge Vertices 1

1 On the **Geometry** toolbar, from the **Virtual Operations** menu, select **Merge Vertices** ( ).

2 Select Point 2.

3 Go to the Merge Vertices settings window. Locate the **Vertex to Remove** section and select Point 3.

4 Click the **Build Selected** button (▦).

MESH 1

In the **Model Builder**, click **Mesh 1** and on the settings window, click **Build All** (▦).

8

Meshing

This chapter describes meshing capabilities, meshing techniques, and meshing operations available for meshing the geometry.

Creating a Mesh for Analysis

Meshing Concepts

The **Mesh** ( ) nodes enable the discretization of the geometry into small units of simple shapes, referred to as *mesh elements*.

A mesh is the result of building a meshing sequence. A meshing sequence corresponding to a geometry consists of Meshing Operations and Attributes. The attribute nodes store properties that are used by the operation nodes when creating the mesh.

Building an operation node creates or modifies the mesh on the part of the geometry defined by the operation node's selection. Some of the operation nodes use properties defined by attribute nodes; for example, the Free Tetrahedral node reads properties from the Distribution and Size attribute nodes. If you choose to import a mesh you have access to a different set of operations (see Operations on Imported Meshes).

For some operation nodes, you can right-click to add local attribute nodes as subnodes. Properties defined in local attribute nodes of an operation node override the corresponding properties defined in global attribute nodes (on the same selection).

GLOBAL VS. LOCAL ATTRIBUTE NODES

An attribute node contains properties defined on a selection. You can add an attribute as a node in the meshing sequence (this is referred to as a *global attribute node*) or add it as a node under an operation node (a *local attribute node*). Global attribute nodes are used by subsequent operation nodes when building the meshing sequence. Local attribute nodes are only used by the owning operation node.

VISUALIZING THE MESH

The Graphics Window shows the resulting mesh from the nodes that have been built. The result of subsequent nodes is not visible. The last built node becomes the *current node* and appears with a quadratic frame around the node's icon ( ). The frame is green if the node and all preceding nodes are built; that is, the mesh in the Graphics window is up to date. The frame is yellow if the node or some preceding node has been edited since the node was built and needs to be rebuilt ( ).

Mesh Elements for 1D, 2D, and 3D Geometries

Virtual Geometry and Mesh Control Operations

1D GEOMETRIES

The mesh generator discretizes the domains (intervals) into smaller intervals (or mesh elements). The endpoints of the mesh elements are called *mesh vertices*.

The boundaries (or vertices) defined in the geometry are represented in the mesh by *boundary elements* (or *vertex elements*).

2D GEOMETRIES

The mesh generator discretizes the domains into *triangular* or *quadrilateral* mesh elements. If the boundary is curved, these elements represent an approximation of the original geometry. The sides of the triangles and quadrilaterals are called *mesh edges*, and their corners are *mesh vertices*. A mesh edge must not contain mesh vertices in its interior.

The boundaries defined in the geometry are discretized (approximately) into mesh edges, referred to as boundary elements (or *edge elements*), which must conform with the mesh elements of the adjacent domains.

The geometry vertices are represented by vertex elements.

3D GEOMETRIES

The mesh generator discretizes the domains into *tetrahedral*, *hexahedral*, *prism*, or *pyramid* mesh elements whose faces, edges, and corners are called *mesh faces*, *mesh edges*, and *mesh vertices*, respectively.

For 3D meshing, platforms handle floating-point operations differently, which sometimes results in slight differences between identical model files that are generated on two different computers.

The boundaries in the geometry are discretized into triangular or quadrilateral boundary elements. The geometry edges are discretized into edge elements.

Similar to 2D, the geometry vertices are represented by vertex elements.

PYRAMID ELEMENTS

Pyramid elements appear in the mesh in these situations. If you:

- Import a NASTRAN® file containing pyramid elements.
- Create a Swept mesh where the source and destination faces share an edge and the source face contains a triangular mesh.
- Convert a quad mesh to a triangular mesh for a face adjacent to a domain that contains a mesh.
- Create a Boundary Layers mesh in 3D for a geometry with sharp edges.

Free (Unstructured) Meshing

The free mesher is available in all space dimensions, and you can use it for all types of geometries regardless of the topology or shape. If you have not defined or generated a mesh, the free mesher automatically creates an unstructured mesh and adds a corresponding node to the Model Builder when a study is computed.

When the free mesher is used:

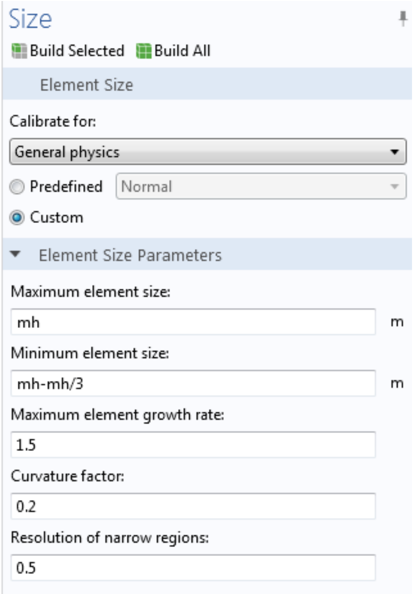
- The number of mesh elements is determined by the shape of the geometry and various mesh parameters.
- The mesh parameters for the free mesher are controlled by Size and Distribution nodes in the meshing sequences.
- You can also control the size of the mesh generated by a specific Free Triangular, Free Quad, or Free Tetrahedral node by adding Size or Distribution subnodes.

2D MODEL UNSTRUCTURED MESHES

Free meshing generates an unstructured mesh with triangular or quadrilateral elements. Use the free mesher in 2D to create an unstructured mesh with triangular or quadrilateral elements. You can combine triangular and quadrilateral meshes by adding domains to the Domain list in the corresponding mesh operation's settings. From here, you can define specific meshing operations to each domain in your model.

3D UNSTRUCTURED MESHES

Free meshing generates an unstructured mesh with tetrahedral elements.



Size

Element Size

Calibrate for:

General physics

○ Predefined Normal

● Custom

Element Size Parameters

Maximum element size:

mh m

Minimum element size:

mh-mh/3 m

Maximum element growth rate:

1.5

Curvature factor:

0.2

Resolution of narrow regions:

0.5

Figure 8-1: An example of custom element mesh sizes. You can also select Predefined element sizes.

Structured Meshes

2D STRUCTURED MESHES

- You can create a structured triangular mesh by using the Convert operation to introduce a diagonal edge to quadrilateral elements.
- Also see 2D and 3D Boundary Layer Meshes below.
- Mapped meshing generates a structured mesh with quadrilateral elements.

  Compared to an unstructured mesh, the interior mesh vertices in a structured mesh are adjacent to the same number of elements. If you want to use mapped meshing on a geometry, you must build the geometry so that the domains are reasonably "regular" in shape and do not contain holes.

3D STRUCTURED MESHES

- Swept meshing generates a structured mesh (at least in the direction of the sweep) with prism or hexahedral elements. See About Swept Meshes.
- Boundary Layers meshing generates structured layers of elements along specific boundaries integrating into an existing structured or unstructured mesh.

2D AND 3D BOUNDARY LAYER MESHES

The meshing type Boundary Layers is an example of a structured mesh. A boundary layer mesh is a mesh with dense element distribution in the normal direction along specific boundaries. This type of mesh is typically used for fluid flow problems in order to resolve the thin boundary layers along the no-slip boundaries.

- In 2D, a layered quadrilateral mesh is used along the specified no-slip boundaries.
- In 3D, the boundary layer mesh is a layered prism mesh or a hexahedral mesh, depending on whether the corresponding boundary-layer boundaries contain a triangular or quadrilateral mesh.

Boundary layer meshes can also resolve large temperature gradients close to heated surfaces subjected to sudden changes over time.

If you have the CFD Module or the Heat Transfer Module, the Heat Sink shows the introduction of a boundary layer mesh at the surfaces of the inner half-circle arc.

About Swept Meshes

A Swept mesh is an example of a semistructured mesh since it is structured in the sweep direction and can be either structured or unstructured orthogonally to the sweep direction. The swept mesher operates on a 3D domain by meshing or reusing an existing mesh on a source face, and then sweeping the resulting face mesh along the domain to an opposite destination face.

You can use several connected faces as source faces. Also the destination can consist of several faces, as long as each destination face corresponds to at least one source face and each source face corresponds to exactly one destination face. Each face about a domain that is to be operated on by the swept mesher is classified as either a source face, a destination face, or a linking face. The linking faces are the faces linking the source and destination faces (see Figure 8-2). The swept mesher can handle domains with multiple linking faces in the sweep direction.

The linking edges are the edges, or the chains of edges, connecting the source and destination faces. For a domain to be possible to sweep, there must be at least one linking edge or chain of edges.

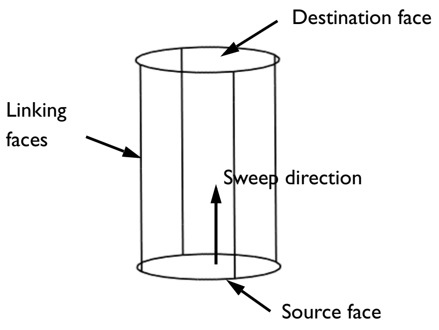


Figure 8-2: Classification of the boundaries about a domain used for swept meshing.

You can specify the source and destination faces manually, but in most cases the swept mesher can automatically identify source and destination faces from the geometry.

- If the source faces are not meshed prior to the sweeping operation the Swept node automatically creates a quadrilateral (or triangular) mesh before sweeping it to the destination.
- If the source faces contain a triangular mesh the resulting swept mesh consists of prism elements.
- If the source faces contain a quadrilateral mesh the resulting swept mesh consists of hexahedral elements.

The default is to create a quadrilateral face mesh but, depending on the source faces, that is not always possible.

For the sweeping technique to work, the geometry must satisfy these criteria:

- Each domain must be bounded by one shell; that is, a domain must not contain holes except if they penetrate both the source and destination face.
- The source and destination for a domain must be opposite each another in the domain's topology.
- Each destination face must correspond to one or more source faces.
- Each source face must correspond to precisely one destination face or a subset of it.
- The cross section along the direction of the sweep for a domain must be topologically constant.

Coincident source and destination faces are allowed.

If any of the faces about a domain is meshed prior to the sweeping operation, the following must be fulfilled.

- If the source and destination faces are meshed, these meshes must match.
- Structured quad meshes must be applied to the linking faces.

Swept and Generating a 3D Swept Mesh

Mesh Control Entities

Sometimes it is desirable to use certain geometric entities only to control the mesh. For example, you can add a curve inside a domain to control mesh element size there. If you mark this curve as a mesh control entity, it is not included in the geometry used when defining the physics and materials. An advantage is that the final mesh need not respect this curve exactly; it is used only to control element size.

Another situation where mesh control entities are useful is when you need precise control of mesh in certain regions of the geometry. In these regions you typically use a structured mesh with distribution nodes to control the mesh. In other regions of the geometry you can use free (unstructured) mesh.

Suppose that you also want to insert boundary layers. If the boundaries separating the domains with structured and free mesh are ordinary geometry boundaries, the boundary layers have to respect them. This can lead to various problems, including low-quality elements or even meshing failures. If you instead mark such boundaries as mesh control entities, the boundary layer mesh algorithm has more freedom to move mesh nodes and to construct a better mesh.

- Using Mesh Control Entities to Control Element Size
- Working with Geometric Entities
- Named Selections

The Mesh Toolbar

- The COMSOL Desktop
- Clear or Delete a Mesh or Solution
- Errors and Warnings
- Meshing Operations and Attributes

After a **Mesh** has been added to the Component node, the **Mesh** ribbon toolbar (Windows) or the **Mesh** contextual toolbar (Mac and Linux) is made accessible. Click a meshing sequence node in the **Model Builder** and the **Mesh** tab or toolbar displays on the COMSOL Desktop.

In general, the instructions throughout the documentation indicate that you click a button on a **Mesh** toolbar, no matter what operating system you are running.

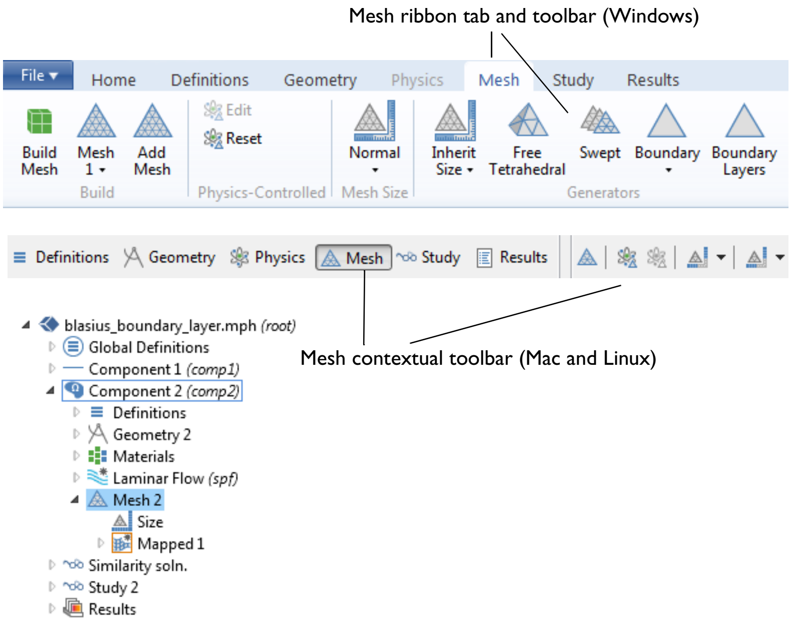


Figure 8-3: Click a Mesh node or any node under the Mesh node (Free Tetrahedral, for example) to open the Mesh ribbon toolbar (Windows users, top) or Mesh contextual toolbar (Mac and Linux users, bottom). Only some of the available buttons are shown in this image.

Click the associated menu arrow and choose the menu item corresponding to the mesh node. If you click the button associated with this menu button COMSOL Multiphysics selects the node in the **Model Builder** corresponding to the last selected item in the menu associated with the menu button.

PREDEFINED MESH ELEMENT SIZES

The following table shows the icons on the drop-down menus for selecting predefined mesh element sizes:

TABLE 8-1: PREDEFINED ELEMENT SIZE ICONS

ICON	NAME	ICON	NAME
△	Extremely Fine	△	Normal
△	Extra Fine	△	Coarse
△	Finer	△	Coarser
△	Fine	△	Extra Coarse
		△	Extremely Coarse

GEOMETRIC MEASUREMENTS

To measure the volume, area, or length of a selected domain, face, or edge, respectively, click the **Measure** button (📏). The result displays in The Messages Window. Using this button it is also possible to view the coordinates of a vertex, the distance between two vertices, or the number of entities and the geometry representation (only if you have license for the CAD Import Module) of a geometry object.

The Mesh Toolbar in 1D and 2D contains a subset of the tools in the 3D toolbar.

Adding, Editing, and Building Meshing Sequences

When you add a new Component to the Model Builder, a meshing sequence is added by default in the node **Mesh 1**. You can add more meshing sequences to the Component by right-clicking the **Component** node and selecting **Mesh**. When a Component has more than one meshing sequence, they are collected under a **Meshes** node. To add new meshing sequences to such a Component, you can alternatively right-click **Meshes**.

You create a mesh by building a meshing sequence, which contains a number of meshing operations as nodes in the sequence. For the default physics-controlled meshes, the software sets up the meshing sequences automatically.

ADDING MESH NODES

Right-click a **Mesh** node (△) in the Model Builder and then select an option from the context menu. Enter the properties in the settings window. In numerical fields you can enter expressions that contain global parameters.

BUILDING SELECTED MESH NODES

To build all nodes (if needed) from the first up to the selected node:

- In a settings window, click the **Build Selected** button (▦).
- Right-click a node under the main **Mesh** node and select **Build Selected**. Or click the node and press F7.

- Windows users: On the **Home** or **Mesh** ribbon toolbars, click **Build Mesh**.
- Mac and Linux users: On the **Model Toolbar** or the **Mesh Contextual Toolbar**, click **Build Mesh**.

BUILD ALL MESH NODES

To build all nodes in the meshing sequence (if needed):

- In a settings window, click the **Build All** button (🟩).
- Right-click the main **Mesh** node (△) and select **Build All**. Or click the **Mesh** node and press F8.
- Windows users: On the **Home** or **Mesh** ribbon toolbars, click **Build Mesh**.
- Mac and Linux users: On the **Model Toolbar** or the **Mesh Contextual Toolbar**, click **Build Mesh**.

EDITING MESH NODES

To edit a mesh node, select it in the tree, and make changes in the settings window that appears. Nodes that you have edited display with an asterisk (*) at the upper-right corner of the icon (△*) in the **Model Builder**. The following nodes are marked with a yellow triangle at the lower-right corner of the node's icon to indicate that they need to be rebuilt. To see the result of your edits in the graphics, use the methods described above to either **Build Selected** or **Build All** nodes.

Using Meshing Sequences: model library path
COMSOL_Multiphysics/Meshing_Tutorials/meshing_sequence

The Mesh Statistics Window

For statistical information about the mesh element quality, right-click the **Mesh** node (△) and select **Statistics** (△). The **Statistics** window includes information about the minimum and average mesh element quality and a mesh element quality histogram, which shows the relative frequency of mesh elements with different quality values. The window contains the following sections:

GEOMETRIC SCOPE

Define the geometric entities for which you want to display the statistics. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to view statistics for the entire mesh.
- Choose **Domain** to specify the domains for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains**.
- Choose **Boundary** to specify the boundaries for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the boundaries in the Graphics window or choose **All boundaries**.
- Choose **Edge** to specify the edges for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the edges in the Graphics window or choose **All edges**. This option is only available in 3D.
- Choose **Point** to specify the points for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the points in the Graphics window or choose **All points**. This option is only available in 2D and 3D.

STATISTICS

In this section you find information on the status of the mesh: for example, if the mesh is empty, if the geometry is partially meshed, or if the geometry is completely meshed. You can select the element type for which you want to see statistics from the **Element type** list. The default is to display statistics for **All elements**.

ELEMENT QUALITY HISTOGRAM

This section displays a histogram plot of the mesh element quality for the specified element type and selection. The x-axis represents the element quality, and the y-axis represents the number of elements of similar quality. The absolute value of the mesh element quality is always between 0 and 1, where 0.0 represents a degenerated element and 1.0 represents the best possible element.

You can also create a histogram plot of the mesh element quality over the total area or volume of the elements by adding a Histogram plot to a 1D Plot Group and using the **Mesh quality** (`qual`) as the expression.

Quality of Elements in the *COMSOL API Reference Manual*

Meshing Techniques

- Choosing a Meshing Sequence Type
- Mesh Element Quality and Size
- Using Several Meshing Sequences of Imported Mesh Type
- Avoiding Inverted Mesh Elements
- Troubleshooting Boundary Layer Mesh Generation
- Troubleshooting Free Tetrahedral Mesh Generation

- Selecting an Element Type
- Errors and Warnings
- Analyzing Model Convergence and Accuracy
- Achieving Convergence When Solving Nonlinear Equations
- Avoiding Strong Transients
- Physics-Related Checks and Guidelines
- Weak Constraints
- Numerical Stabilization
- Weak Contribution (PDEs) and Weak Contribution (ODEs and DAEs)
- Geometric Variables and Mesh Variables

Choosing a Meshing Sequence Type

PHYSICS-CONTROLLED MESH

If you select **Physics-controlled mesh** in the **Sequence type** list in the settings window of a mesh node and build the meshing sequence, COMSOL Multiphysics creates a mesh that is adapted to the current physics settings in the model. The default is to use physics-controlled mesh. For example, for a fluid-flow model you get a somewhat finer mesh than the default with a boundary layer mesh along the no-slip boundaries. If you want to modify the overall element size of the physics-induced mesh you select another element size from the **Element size** list in the settings window of the main mesh node and rebuild the mesh. If you change the physics settings in the model and rebuild the meshing sequence, COMSOL creates a new mesh adapted to the new physics settings.

A physics-induced mesh is not adapted by numerical error estimates—that type of adaptive meshing is provided by mesh adaptation in the solver sequence.

To edit a physics-induced meshing sequence, or to see the errors and warnings of a failing mesh build, select **User-controlled mesh** in the **Sequence type** list or right-click the Mesh node and select **Edit Physics-Induced Sequence** ( ). The program then adds the nodes under the main **Mesh** node that together form the physics-controlled mesh.

By doing this the sequence is no longer updated according to changes that applied to the physics settings in the model.

If you right-click the **Mesh** node and select **Reset to the Physics-Induced Sequence** ( ), the sequence is reset to the physics-induced sequence. However, the type of the sequence is still **User-controlled mesh**. To switch back to physics-controlled meshing, select **Physics-controlled mesh** in the **Sequence type** list in the settings window of the mesh node. If you add a node to the sequence, the type of the sequence automatically switches to **User-controlled mesh**.

USER-CONTROLLED MESH

Alternatively, you can use a user-controlled mesh. It is then possible to manually build and edit the meshing sequence using the meshing techniques described below for creating a 2D and 3D meshes.

If you select **User-controlled mesh** from the **Sequence type** list in the main **Mesh** node's settings window, the program adds a **Size** node and a node for the default mesher (**Free Triangular** in 2D, for example). If the **Sequence type** list is set to **Physics-controlled** mesh and you add a **Size** node, for example, the mesh sequence switches to a user-controlled mesh, but no default mesher is added.

Mesh Element Quality and Size

The mesh resolution and mesh element quality are important aspects to consider when validating a model. Low mesh resolution—in relation to the variations in the solution and the geometry—can lead to inaccurate results, and a low mesh element quality—which measures the regularity of the mesh elements' shapes—can lead to inverted mesh elements (see Avoiding Inverted Mesh Elements) and to high condition numbers for the Jacobians, which in turn can cause convergence issues.

COMSOL Multiphysics includes built-in variables for these mesh quantities:

- `h`, the local mesh size
- `qual`, the mesh element quality, which is a dimensionless quantity between 0 and 1, where 1 represents a perfectly regular element, and 0 represents a degenerated element.

DISPLAYING MESH ELEMENT QUALITY AND THE MESH ELEMENT SIZE

You can display the mesh element quality and the mesh element size using, for example, a surface plot in 2D or a volume plot in 3D. You can always use a Mesh data set to display these quantities as soon as you have created a mesh. If you have a solution, you can also use a Solution data set. For a Component with a mesh, do the following steps to display the mesh element quality or mesh element size:

1. Right-click the **Mesh** node and select **Plot** ( ). This creates a plot group with a **Mesh** plot node ( ).
2. By default, this plot shows the mesh element quality. In the **Mesh** settings window, select **Size** instead of **Quality** from the **Element color** list to plot the mesh element size instead.

Alternatively, you can access the built-in variables for mesh element quality (`qual`) and mesh element size (`h`) in a surface plot, for example:

1. Under **Results** ( ), right-click **Data Sets** ( ) and select **Mesh** ( ).

2 Add a 2D or 3D Plot Group using the Mesh data set as the group's data set, and then add a Surface or Volume plot. For example, in a **2D Plot Group>Surface** node (▣), select **Element size (h)** or **Element quality (qual)** from the predefined quantities (under **Mesh**). Then click the **Plot** button (▣).

- Mesh (Data Set)
- Mesh (Plot)
- Mesh (Export)
- Mesh Report Node

Avoiding Inverted Mesh Elements

INVERTED MESH ELEMENTS

If you have a mesh that is coarse along a curved boundary, you might encounter problems with inverted mesh elements. This means that a mesh element is wrapped inside-out or has zero area (in 2D) or volume (in 3D). More precisely, there is some coordinate for which the Jacobian matrix for the mapping from local to global coordinates has a negative or zero determinant. In most cases, the linear (straight) mesh elements that you see in a mesh plot are not inverted, but the higher-order curved mesh elements used for computing the solution might be. Studying the minimum element quality therefore does not reveal the presence of inverted mesh elements in most cases.

Inverted mesh elements in themselves do not pose any immediate threat to the overall accuracy of your solution. However, if you are using an iterative solver, it might fail to converge. If you reach convergence and the solution looks good, it likely is. It is worth bearing in mind that the faces where there are inverted elements are less than perfectly resolved. If these faces are important for your results, you might want to pursue a mesh without inverted elements or at least make sure that the mesh resolution is sufficiently fine to guarantee an accurate solution. The easiest way to get an idea of the accuracy is to try a few different meshes and see how the solution changes. If the variation does not exceed your limits of acceptance, you are fine.

The solver prints warnings about inverted mesh elements to the **Log** window (▤) if they appear. Warning nodes (⚠) also appear in the solver sequence where the inverted mesh elements appear. It is often possible to avoid problems with inverted mesh elements by reducing the geometry shape order or by modifying the geometry or the mesh.

USING LINEAR GEOMETRY SHAPE ORDER

When solving a model, the solver ensures that no inverted mesh elements are created. This is done by reducing the geometry shape order for the corresponding elements to first order. By default, the solver does this automatically. Alternatively, you can avoid problems with inverted mesh elements by using linear geometry shape order for all elements. You do this by choosing **Linear** from the **Geometry shape order** list in the **Model Settings** section of the settings window for the main **Component** node.

VISUALIZING LINEARIZED ELEMENTS

The variable `linearizedelem` is 1 in elements that are linearized and 0 elsewhere. You can use this variable to identify mesh elements with linearized elements.

MODIFYING THE GEOMETRY OR MESH

If you do not want to use linear geometry shape order to avoid problems with inverted mesh elements you can try any of the following:

- Create a swept 3D mesh instead of using the free mesher.
- Avoid small curved boundaries such as fillets unless they are important for the result.

VISUALIZING INVERTED MESH ELEMENTS

You can visualize inverted mesh elements using the built-in `reldetjacmin` variable, which is the minimum of the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates. A minimum value less than zero for an element means that the element is wrapped inside-out; that is, it is an inverted mesh element.

A typical visualization uses `reldetjacmin` as the quantity to plot as a volume plot. To display only the inverted elements, add a **Filter** subnode using the logical expression `reldetjacmin<0` to include only the inverted elements.

- Adaptive Mesh Refinement (attribute node)
- Automatic Remeshing
- The Progress Window
- The Log Window
- Adaptive Mesh Refinement (Utility Node)
- Meshing Sequence

Troubleshooting Boundary Layer Mesh Generation

The boundary layer meshing algorithm is sensitive to the topology of the model geometry. If you get an error when trying to build a Boundary Layers node, try the following:

- Remove unnecessary interior boundaries such as boundaries (resulting from Boolean operations of geometry objects) that do not separate materials or physics. An efficient way to do this is to mark these boundaries as Mesh Control Entities in the Geometry Sequence. Once you have removed unnecessary boundaries, mesh the domains using a Free Tetrahedral or a Swept Mesh. When the domains are meshed, the control boundaries are automatically removed, and you can insert boundary layers, ignoring the interfering boundaries.
- Use boundary layer mesh trimming instead of splitting. By default, the boundary layer mesher creates a boundary layer split at each sharp corner in 2D and along each sharp edge in 3D.

To turn off boundary layer splits, see Boundary Layers.

Troubleshooting Free Tetrahedral Mesh Generation

This section gives you some suggestions about how to solve problems that you might encounter when creating tetrahedral meshes.

BUILD A FINER MESH

As a general rule, it is easier to construct a mesh with smaller elements than a mesh with larger elements. If you get errors or low-quality elements when you try to mesh certain domains, try to decrease the element size using appropriate Size attributes on these domains.

USE AN APPROPRIATE MINIMAL ELEMENT SIZE

If your geometry contains details that are very small compared to the total volume of the mesh, you must ensure that the Minimum element size parameter in the corresponding Size attribute is at least as small as the smallest detail you wish to resolve. If this parameter is too large, you get warnings when building the node. For example, the warning “Edge is much shorter than the specified minimum element size” indicates that there are edges significantly shorter than the specified minimum element size. The resulting mesh gets badly shaped elements.

💡 To locate small details, such as short edges and sliver faces, you can add and build a Free Tetrahedral node with normal size settings. Doing this results in warnings with selections that point you to the corresponding small entities. You can also inspect the mesh visually to locate unexpected small elements.

REMOVE UNWANTED GEOMETRY DETAILS

Sometimes, the geometry contains small features, like sliver faces and short edges, which you do not wish to resolve at all. Then you can use Virtual Geometry Operations in the sequence to ignore disturbing details of the geometry.

If you have a license for the CAD Import Module, you can also use CAD defeaturing operations to simplify the geometry.

PARTITION THE GEOMETRY INTO SIMPLE DOMAINS

If the geometry has very complex domains or very complex faces that you have trouble meshing, you can try to partition the geometry into less complex entities. On a philosophical level, this method could be classified as a “divide and conquer” strategy. It is often possible the Partition geometry feature to partition a complex domain into two domains. You can use a Mesh Control Faces node to make this partitioning only when building the mesh (see Mesh Control Entities).

💡 To split a solid geometry object into parts using a Work Plane, place the work plane where you want to cut the domain. Then add a **Partition** node from the **Boolean Operations** submenu, and select **Work plane** from the **Partition with** list in the **Partition** node’s settings window.

Meshing Operations and Attributes

A meshing sequence corresponding to a geometry consists of Mesh Operations and Mesh Attributes. The attribute nodes store properties that are used by the operation nodes when creating the mesh. You can also choose the Predefined Mesh Element Sizes.

MESH OPERATIONS

The following table lists the available mesh operations:

TABLE 8-2: OPERATION NODES

ICON	NAME	DESCRIPTION	GEOMETRIC ENTITY LEVEL
	Boundary Layers	Create a boundary layer mesh—a mesh with dense element distribution in the normal direction along specific boundaries. This mesh is typically used for fluid flow problems to resolve the thin boundary layers along the no-slip boundaries.	Domain
	Convert	Some geometries have domains that are well suited for swept meshes. If there are surrounding domains that cannot be swept, you can use this to convert faces with quadrilateral mesh between these domains to faces with triangular mesh.	Domain and boundary
2D 3D	Copy Domain	Copy a mesh between domains,	Domain
2D 3D	Copy Edge	Copy a mesh between edges. A 3D Copy Edge feature can also be used for destination edges of different shapes.	Edge
	Copy Face	To make a copy of a mesh that you can use to create an identical mesh on, for example, two boundaries in a model with periodic boundary conditions.	Boundary
	Edge	To create an edge mesh. This menu button works in the same way as the Free Tetrahedral button.	Edge
	Free Quad	Create an unstructured quadrilateral mesh. This menu button works in the same way as the Free Tetrahedral button.	Boundary
	Free Tetrahedral	Create an unstructured tetrahedral mesh. If no selection is specified, this feature creates a mesh on the remaining domains, boundaries, edges and points.	Domain, boundary, edge, or point
	Free Triangular	Create an unstructured triangular mesh. This menu button works in the same way as the Free Tetrahedral button.	Boundary

TABLE 8-2: OPERATION NODES

ICON	NAME	DESCRIPTION	GEOMETRIC ENTITY LEVEL
	Mapped	To create a structured quadrilateral mesh on boundaries in 3D and domains in 2D.This menu button works in the same way as the Free Tetrahedral button.	Boundary
	Reference	To refer to another meshing sequence. Building a Reference node runs the operation nodes of the referenced sequence.	
	Refine	Refine a mesh by splitting elements.	
	Swept	Create a swept mesh. In domain selection mode this button works in the same way as the Free Tetrahedral button. In boundary selection mode the software creates a swept mesh on the remaining domains using the selected boundaries as source faces.	Domain

MESH ATTRIBUTES

The following table lists the mesh attributes:

TABLE 8-3: MESH ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
	Boundary Layer Properties	To specify the location of the boundary layers and the properties, such as the number and thickness of the boundary layers.
	Corner Refinement	To decrease the element size at sharp corners. The node considers a vertex in 2D or an edge in 3D to be a sharp corner if the angle between the adjacent selected boundaries, with respect to the selected domain, is greater than a specified angle.
	Distribution	To specify the distribution of mesh elements along an edge, for example. It is possible to add Distribution nodes both as global nodes and as local nodes.
	Edge Groups	To specify the four groups of edges around a boundary (3D) or domain (2D) that is used to determine the Mapped mesh of the boundary/domain.
	Edge Map	Using this node, the source mesh of the Copy Face or Copy Domain operation is transformed so that the source edge of the Edge Map node is mapped onto the destination edge of the Edge Map node with the specified orientation.
	One-Point Map	To define the orientation of the source mesh on the destination for a Copy Face or Copy Domain feature by specifying how to map one point adjacent to the source to a point adjacent to the destination.
	Scale	To scale the properties of the Size, Distribution, and Boundary Layer Properties nodes. I

TABLE 8-3: MESH ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
	Size	To specify the size of mesh elements. It is possible to add Size nodes both as global nodes and as local nodes.
	Two-Point Map	To define the orientation of the source mesh on the destination for a Copy Face or Copy Domain feature by specifying how to map a pair of points adjacent to the source to a pair of points adjacent to the destination.

Boundary Layers

A **Boundary Layers** mesh ( ) is a mesh with dense element distribution in the normal direction along specific boundaries. This type of mesh is typically used for fluid flow problems to resolve the thin boundary layers along the no-slip boundaries.

In 2D, a layered quadrilateral mesh is used along the specified no-slip boundaries.

In 3D, the boundary layer mesh is a layered prism mesh or a hexahedral mesh depending on whether the corresponding boundary-layer boundaries contain a triangular or a quadrilateral mesh.

Additional elements of an arbitrary type can also be inserted into the layers if needed.

To create a boundary layer mesh:

- In the **Graphics** window, select the boundaries where you want to insert boundary layer elements. On The Mesh Toolbar click the **Boundary Layers** button ( ). This adds a node with the same name, as well as a default **Boundary Layer Properties** node and at the same time inserts boundary layer elements for the selected boundaries, with the meshed domains adjacent to the selected boundaries as domain selection.
- On the **Mesh** ribbon toolbar (Windows) or from the **Mesh** contextual toolbar (Mac and Linux) click the **Boundary Layers** button ( ).
- Right-click a 2D or 3D **Mesh** node and select **Boundary Layers**.

Then enter the properties for the boundary layer mesher using the following sections:

DOMAIN SELECTION

Specify the domains where you want a boundary layer mesh by first choosing an option from the **Geometric entity level** list:

- Choose **Entire geometry** to specify boundary layer mesh for the entire geometry.
- Choose **Domain** to specify the domains for which you want a boundary layer mesh. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains** to select all domains.

CORNER SETTINGS

The following options for handling boundary layers at sharp corners are available from the **Handling of sharp corners** list (in 2D) and the **Handling of sharp edges** list (3D):

- Select **Splitting** (the default) to introduce boundary layer splits at sharp corners. In the **Minimum angle for splitting** field you specify the minimum angle between adjacent boundary layer boundaries for a split to occur. Control the maximum angle of the elements in the split region by the **Maximum angle per split** parameter.
- Select **Trimming** to trim the boundary layer mesh at sharp corners. In the **Minimum angle for trimming** and in the **Maximum angle for trimming** fields you specify the minimum angle and maximum angle, respectively, between adjacent boundary layer boundaries for trimming to occur.
- Select **None** to not use any special treatment at sharp corners.

In the **Maximum layer decrement** field you can specify the maximum difference in number of boundary layers between neighboring points on boundary layer boundaries.

TRANSITION

Select the **Smooth transition to interior mesh** check box to smooth the transition in element size from the boundary layer mesh to the interior mesh. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

When a **Boundary Layers** node is added, a **Boundary Layer Properties** node is automatically added as a subnode. Use this subnode to specify the boundary layers and the properties of the boundary layers. If you want to specify different boundary layer properties for more than one boundary selection, right-click the **Boundary Layers** node and add additional **Boundary Layer Properties** subnodes. However, adjacent boundaries must have the same number of boundary layers.

- If you have the Batteries & Fuel Cells Module, see Soluble Lead-Acid Redox Flow Battery (2D): model library path **Batteries_and_Fuel_Cells_Module/Batteries/pb_flow_battery**.
- If you have the CFD Module, see Turbulent Flow Over a Backward Facing Step (2D): model library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**.
- If you have the Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: model library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**.

Boundary Layer Properties

Boundary Layer Properties

Add a **Boundary Layer Properties** node ( ) to specify the location of the boundary layers and the properties, such as the number and thickness of the boundary layers.

By default one Boundary Layer Properties node is added as a subnode to a Boundary Layers node. To add additional nodes, right-click a 2D or 3D **Mesh** node and select **Boundary Layers Properties**; then enter the properties using the following sections:

BOUNDARIES

Define the boundaries where you want boundary layers. Choose **Manual** in the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.

BOUNDARY LAYER PROPERTIES

The default **Number of boundary layers** is 8.

The default **Boundary layer stretching factor** is 1.2. This field is used to specify the increase in thickness between two consecutive boundary layers as a scaling factor; for example, entering `1.3` means that the thickness increases by 30% from one layer to the next.

To specify the thickness of the first element layer—the layer adjacent to the corresponding boundary—choose an option from the **Thickness of first layer** list—**Automatic** (the default) or **Manual**.

- If **Automatic** is kept as the default, the thickness of the first layer is 1/20 of the local domain element height. Enter the **Thickness adjustment factor** to specify a scaling factor that multiplies this default size. The default is 1.
- If **Manual** is selected, the default **Thickness** (SI unit: m)) is 0.00118 m. Enter another value in the field as required.

!	The boundary layer meshing algorithm shrinks the boundary layers automatically if needed, for example, due to a narrow region. However, the stretching factor is always respected. In some cases the boundary layer meshing algorithm can choose to create fewer layers than specified. If this happens a warning is printed to the Log page of The Progress Window.

Convert

Some geometries have domains that are well suited for swept meshes. If there are surrounding domains that cannot be swept, you can **Convert** (□◪) faces with quadrilateral mesh between these domains to faces with triangular mesh. This makes it possible to generate adjacent-free tetrahedral mesh. Pyramid elements are generated in the interface between the triangular mesh of the converted face and the hexahedral or prism mesh in the domain.

- It is possible to convert the entire mesh to tetrahedral mesh. This is useful because there are a few computations, such as the adaptive solver, that can only be used with a simplex mesh (that is, a mesh with only tetrahedral and triangular elements).
- You can convert a mixed mesh, consisting of tetrahedral, pyramid, prism, and hexahedral elements, to a pure tetrahedral mesh. The mesh conversion splits elements into several tetrahedral elements.
- In 2D, and on faces in 3D, you can convert a mesh with quadrilateral elements to a mesh with only triangular elements.
- In 3D, adjacent domain elements are also split to conform to the split face elements.

To convert a mesh:

- In the **Graphics** window, select the domains or boundaries where you want to convert the elements. Then click the **Convert** button (□◪) on The Mesh Toolbar. This adds a node with the same name and converts the mesh on the selected entities:
  - On the **Mesh** ribbon toolbar (Windows) from the **Operations>Modify** (△) menu, choose **Elements>Convert**.
  - From the **Mesh** contextual toolbar (Mac and Linux), from the **Modify** (△) menu choose **Elements>Convert**.
- Right-click a 2D or 3D **Mesh** node and choose **More Operations>Convert**.

Then use the following sections to specify the parts of the mesh to convert and the method that the conversion uses to split the elements:

GEOMETRIC SCOPE (3D) / DOMAINS (2D)

First define the geometric entities where you want to convert the mesh elements. You choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to convert the mesh elements on all domains (and all boundaries in 3D).
- Choose **Domain** to specify the domains for which you want to convert mesh elements. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Boundary** to specify the boundaries for which you want to convert mesh elements. Choose **Manual** in the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries. This option is only available in 3D.

ELEMENT SPLIT METHOD

From the **Element split method** list, select **Insert diagonal edges** (the default setting) to split each quadrilateral element into two triangular elements and each hexahedral element into five tetrahedral element, or select **Insert center points** to split each quadrilateral element into four triangular elements and each hexahedral element into 28 tetrahedral elements. The conversion also affects quadrilateral elements on the boundaries of the specified domains in 3D.

Both element split methods split each prismatic element into three tetrahedral elements. When pyramid elements are involved in a split, also other splits can be performed.

Corner Refinement

Add a **Corner Refinement** node ( ) to decrease the element size at sharp corners. The node considers a vertex in 2D or an edge in 3D to be a sharp corner if the angle between the adjacent selected boundaries, with respect to the selected domain, is greater than a specified angle.

- On the **Mesh** ribbon toolbar (Windows) from the **Operations>Modify** ( ) menu, choose **Size>Corner Refinement**.
- From the **Mesh** contextual toolbar (Mac and Linux), from the from the **Modify** ( ) menu choose **Size>Corner Refinement**.
- To add it as a global node, right-click a **Mesh** node and select **Corner Refinement**. To add it as a subnode to a Mesh Operations node, right-click an operation node and select **Corner Refinement**. See Global vs. Local Attribute Nodes.

If there are several Corner Refinement nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last Corner Refinement node in the sequence.

Then enter the properties using the following sections:

DOMAIN SELECTION

Specify the domains for which the node determines if the specified corners are sharp. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to specify that the node should determine sharp corners with respect to all domains. The corner refinement also considers corners not adjacent to any domain.
- Choose **Domain** to specify the domains for which you want to determine sharp corners. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

By default Domain Selections are **Active** (the **Active** button is ON).

BOUNDARY SELECTION

To specify the boundaries, click the **Active** button to turn it ON, and select the boundaries in the **Graphics** window for which the corner refinement should determine the sharp corners. When the boundary selection is **Active**, the **Domain Selection** section **Active** button is automatically turned OFF.

ANGLE

Use the **Minimum angle between boundaries** field to specify the minimum angle between a pair of adjacent boundaries in the boundary selection for the refinement factor to apply at the vertex in 2D and edge(s) in 3D between the two boundaries. If a boundary pair is adjacent to one domain on each side (interior boundary) the corner refinement determines the angle(s) on the side(s) corresponding to the specified domain(s).

REFINEMENT

Use the **Element size scaling factor** field to specify a refinement factor (<1) that scales the element size for the vertices in 2D and edges in 3D corresponding to the sharp corners.

Copy Domain

Add a **Copy Domain** node to 2D () or 3D () models to create identical meshes on domains.

Copying a mesh to a destination domain in 2D that is adjacent to a meshed domain is possible if the edges between these domains have the same number of elements as the corresponding source edges. The mesh on the destination edges is kept and the copied domain elements are modified to fit with this edge mesh.

Copying a mesh to a destination domain in 3D that is adjacent to a meshed domain is also possible if each face between these domains has a mesh isomorphic to a mesh of the corresponding source face. The mesh on the destination face is kept and the copied domain elements are modified to fit with this face mesh.

The edges around the source and destination domains in 2D are allowed to be partitioned differently but only in such a way that several edges of the source domain map to one edge of the destination edge. Not the other way around.

The faces around the source and destination domains in 3D are also allowed to be partitioned differently with exactly that same limitation (source to destination face mapping must be many-to-one).

To copy a mesh between domains:

- In the **Graphics** window, select both the domains to copy the mesh from and the domains to copy the mesh to. Then on The Mesh Toolbar click the **Copy Domain** button in 2D () or 3D (). This adds a node with the same name and copies the mesh and includes the source domains set to the selected domains with a mesh and the destination domains set to the selected domains without a mesh.
  - On the **Mesh** ribbon toolbar (Windows) from the **Operations>Copy** () menu, choose **Copy Domain**.
  - From the **Mesh** contextual toolbar (Mac and Linux), from the **Copy** menu (), choose **Copy Domain**.
- Right-click a **Mesh** node and choose **More Operations>Copy Domain**.

Then enter the properties for the copy meshing operation using the following sections:

SOURCE AND DESTINATION DOMAINS

It is possible to copy a mesh from one or several source domains onto one or several destination domains. The source (or their combination, if many-to-one is used) must be a connected set of exactly the same shape as the corresponding destination, up to a constant scaling factor. More precisely, the distance between any two geometry vertices on the destination is required to be the same, up to a constant scaling factor, as the distance between the corresponding geometry vertices on the source.

Click the **Active** button to toggle between turning ON and OFF selections. Select the domains to copy the mesh from in the **Graphics** window. For Windows users, the buttons are ON and OFF. For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

The source domains must be connected when **Single destination (many-to-one)** option is specified. In an assembly, an identity pair is not sufficient to connect boundaries across parts. Instead, consider forming a union of the parts or splitting the destination boundary (using imprints, for example) so that the mesh copy is a one-to-one copy operation using two or more **Copy Domain** nodes.

TYPE OF COPY

Select **Automatic** (the default) to let the software determine the proper copy method, select **Single destination (many-to-one)** to let the entire source mesh be copied onto each destination entity separately, and select **Array copy (many-to-many)** to let each single source entity mesh be copied onto a corresponding single destination entity.

!

Array copy (many-to-many) can be used only if a bijective transformation of source to destination can be found (a transformation that sets 1-to-1 mapping between source and destination).

SWITCH SELECTIONS

Click the **Switch Source and Destination** button to switch source and destination selections. Edge map is available to be switched, if provided.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

To control the orientation of the source mesh on the destination when using the **Copy Domain** node, right-click and add an Edge Map, One-Point Map, or Two-Point Map node as a local attribute.

Copy Edge

Add a **Copy Edge** node to 2D ( ) and 3D ( ) models to create identical meshes on edges. A 3D **Copy Edge** feature can also be used for destination edges of different shapes.

To copy a mesh between edges:

- In the **Graphics** window, select both the edges to copy the mesh from and the edges to copy the mesh to. On The Mesh Toolbar click the **Copy Edge** button for 2D ( ) or 3D ( ) models. This adds a node with the same name and copies the mesh and includes the source edges set to the selected edges with a mesh and the destination edges set to the selected edges without a mesh.
  - On the **Mesh** ribbon toolbar (Windows) from the **Operations>Copy** ( ) menu, choose **Copy Edge**.
  - From the **Mesh** contextual toolbar (Mac and Linux), from the **Copy** menu ( ), choose **Copy Edge**.
- Right-click a **Mesh** node and choose **More Operations>Copy Edge**.

Then enter the properties for the copy meshing operation using the following sections:

SOURCE AND DESTINATION EDGES

It is possible to copy a mesh from one or several source edges onto one or several destination edges. The source (or their combination, if many-to-one is used) must be a connected set of exactly the same shape as the corresponding destination, up to a constant scaling factor. More precisely, the distance between any two geometry vertices on the destination is required to be the same, up to a constant scaling factor, as the distance between the corresponding geometry vertices on the source.

Click the **Active** button to toggle between turning ON and OFF selections. Select the edges to copy the mesh from in the **Graphics** window. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

> The source edges must be connected when **Single destination (many-to-one)** option is specified. In an assembly, an identity pair is not sufficient to connect boundaries across parts. Instead, consider forming a union of the parts or splitting the destination boundary (using imprints, for example) so that the mesh copy is a one-to-one copy operation using two or more **Copy Face** or **Copy Edge** nodes.

TYPE OF COPY

See Copy Domain for settings information.

SWITCH SELECTIONS

See Copy Domain for settings information.

CONTROL ENTITIES

See Copy Domain for settings information.

ORIENTATION

Select **Automatic orientation** to let the software determine the orientation of the source mesh on the destination automatically (this is the default), select **Same orientation** to let the source mesh be copied to the destination according to the direction of the edges, and select **Opposite orientation** to let the source mesh be copied to the destination in the opposite direction. Use the option **Show edge direction arrows** in the **View** node under the **Definitions** node to view the arrow direction.

Copy Face

For 3D models, use a **Copy Face** node ( ) to make a copy of a mesh that you can use to create an identical mesh on, for example, two boundaries in a model with periodic boundary conditions.

Copying a mesh to a face that is adjacent to a meshed face is possible if the edges between these faces have the same number of elements as the corresponding source edges. The mesh on the destination edges is kept and the copied face elements are modified to fit with this edge mesh.

The edges around the source and destination faces are allowed to be partitioned differently, but only in such a way that several edges of the source face map to one edge of the destination edge. Not the other way around.

> Copying a face mesh in 3D is only possible if the destination face is not adjacent to any meshed domain. The copy node overwrites any existing mesh on the destination face.

To copy a mesh between faces:

- In the **Graphics** window, select both the boundaries to copy the mesh from and the boundaries to copy the mesh to. On The Mesh Toolbar click the **Copy Face** () button. This adds a node with the same name and copies the mesh and includes the source boundaries set to the selected boundaries with a mesh and the destination boundaries set to the selected boundaries without a mesh.
  - On the **Mesh** ribbon toolbar (Windows) from the **Operations>Copy** () menu, choose **Copy Face**.
  - From the **Mesh** contextual toolbar (Mac and Linux), from the **Copy** menu (), choose **Copy Face**.
- Right-click a **Mesh** node choose **More Operations>Copy Face**.

Then enter the properties for the copy meshing operation using the following sections:

SOURCE AND DESTINATION BOUNDARIES

It is possible to copy a mesh from one or several source boundaries onto one or several destination boundaries. The source (or their combination, if many-to-one is used) must be a connected set of exactly the same shape as the corresponding destination, up to a constant scaling factor. More precisely, the distance between any two geometry vertices on the destination is required to be the same, up to a constant scaling factor, as the distance between the corresponding geometry vertices on the source.

Click the **Active** button to toggle between turning ON and OFF selections. Select the boundaries to copy the mesh from in the **Graphics** window. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

> The source boundaries must be connected when **Single destination (many-to-one)** option is specified as the **Type of Copy**. In an assembly, an identity pair is not sufficient to connect boundaries across parts. Instead, consider forming a union of the parts or splitting the destination boundary (using imprints, for example) so that the mesh copy is a one-to-one copy operation using two or more **Copy Face** nodes.

TYPE OF COPY

See Copy Domain for settings information.

SWITCH SELECTIONS

See Copy Domain for settings information.

CONTROL ENTITIES

See Copy Domain for settings information.

> The Copy Edge feature has an orientation section. To control the orientation of the source mesh on the destination when using the **Copy Face** node, right-click and add an Edge Map, One-Point Map, or Two-Point Map node as a local attribute.

Distribution

Use the **Distribution** node (▦) to specify the distribution of mesh elements along an edge, for example. It is possible to add **Distribution** nodes both as global nodes and as local nodes. If there are several **Distribution** nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last **Distribution** node in the sequence. Distribution properties always override properties defined by **Size** nodes sharing the same selections.

- On the **Mesh** ribbon toolbar (Windows) from the **Operations>Modify** (▲) menu, choose **Size>Distribution**.
- From the **Mesh** contextual toolbar (Mac and Linux), from the **Modify** (▲) menu, choose **Size>Distribution**.
- To add it as a global node, right-click a **Mesh** node and select **Distribution**. To add it as a subnode to an operation node, right-click a Mesh Operations node and select **Distribution**. See Global vs. Local Attribute Nodes.

GEOMETRIC SCOPE (3D) / BOUNDARIES (2D) / DOMAIN SELECTION (1D)

Define the geometric entities where you want to specify a distribution. Choose the level of the geometry from the **Geometric entity level** list (only available in 3D):

- Choose **Domain** to specify the domains for the distribution. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Edge** to specify the edges for the distribution. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges.

Edge is the only option in 2D, and **Domain** is the only option in 1D.

DISTRIBUTION

There are three main distribution methods that you select from the **Distribution properties** list:

- Select **Explicit distribution** to use an explicit, user-defined element distribution. To define the distribution of mesh elements, enter a vector-valued expression of strictly increasing values starting with zero (using comma-separated numbers), specifying the relative arc length values of the mesh vertices along the edge or boundary.
- Select **Fixed number of elements** to use a fixed number of mesh elements, which you enter into the **Number of elements** field. This is the default option.
- Select **Predefined distribution type** to specify properties of a predefined distribution method that can be a geometric sequence (exponentially increasing or decreasing element size) or an arithmetic sequence (equal distance between elements); see below for details.

Predefined Distribution Type Settings

In the **Number of elements** field, enter the number of elements (the default is 5 elements). To specify the ratio in size between the last element and first element in the distribution, use the **Element ratio** field (the default value is 1.0; that is, the first and the last elements have the same size). From the **Distribution method** list, select **Arithmetic sequence** for a linear element distribution or select **Geometric sequence** for an exponential element distribution. Select the **Symmetric** check box to get a symmetric distribution, and select the **Reverse direction** check box to switch the element distribution to the opposite direction along the edge or boundary. If you have specified several edges in the selection the **Reverse direction** check box refers to the edge in the selection with lowest entity number (the *master edge* in the selection). For the other edges, their direction (with respect to the distribution) is such that the rotation with respect to the master edge is minimized.

Meshing Operations and Attributes

Edge

Add an **Edge** node (△) to mesh edges. You can control the number of elements and the distribution of elements in the edge mesh by using Size and Distribution nodes.

To create an edge mesh:

- On the **Mesh** ribbon toolbar (Windows) from the **Generators>Boundary** (△) menu, choose **Edge**.
- From the **Mesh** contextual toolbar (Mac and Linux), **Boundary** menu (△), choose **Edge**.
- Right-click a **Mesh** node and choose **More Operations>Edge**.

Then enter the properties using the following sections:

EDGES (3D) / BOUNDARIES (2D) / DOMAINS (1D)

To define the edges where you want a create a mesh, first choose the level of the geometric entities from the **Geometric entity level** list:

- Choose **Entire geometry** to specify an edge mesh for the entire geometry.
- Choose **Remaining** to specify an edge mesh for remaining, unmeshed edges.
- Choose **Edge** (3D), **Boundary** (2D), or **Domain** (1D) to specify the edges for which you want to create a mesh. Choose **Manual** in the **Selection** list to select the edges in the **Graphics** window or choose **All edges** (3D), **All boundaries** (2D), or **All domains** (1D) to select all edges.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

Edge Groups

Use an **Edge Groups** node (▦) to specify the four groups of edges around a boundary (3D) or domain (2D) that is used to determine the Mapped mesh of the boundary/domain.

For all the settings sections, click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

To add this node, right-click Mapped and select **Edge Groups**. Then enter the properties using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundary/domain where you want to specify the edge groups. Choose **Manual** in the **Selection** list to select the boundary/domain in the **Graphics** window.

FIRST EDGE GROUP

Activate the **First Edge Group** list and select the edges for the first edge group in the **Graphics** window.

SECOND EDGE GROUP

Activate the **Second Edge Group** list and select the edges for the second edge group in the **Graphics** window.

THIRD EDGE GROUP

Activate the **Third Edge Group** list and select the edges for the third edge group in the **Graphics** window.

FOURTH EDGE GROUP

Activate the **Fourth Edge Group** list and select the edges for the fourth edge group in the **Graphics** window.

Edge Map

Use an **Edge Map** node ( ) to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node. Using this node, the source mesh of the Copy Face or Copy Domain operation is transformed so that the source edge of the Edge Map node is mapped onto the destination edge of the Edge Map node with the specified orientation.

EDGES

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

- Activate the **Source edge** list and select the edge that you want to define as source edge in the **Graphics** window.
- Activate the **Destination edge** list and select the edge that you want to define as destination edge in the **Graphics** window.

ORIENTATION

Select **Automatic orientation** to let the software determine the orientation of the mesh of the source edge on the destination edge (this is the default), select **Same orientation** to let the mesh of the source edge be copied to the destination edge according to the directions of the edges, or select **Opposite orientation** to let the mesh of the source edge be copied to the destination edge in the opposite direction.

Free Quad

Add a **Free Quad** node (▦) to create an unstructured quadrilateral mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using Size and Distribution nodes.

To create an unstructured quadrilateral mesh:

- On the **Mesh** ribbon toolbar (Windows) from the **Generators>Boundary** (△) menu, choose **Free Quad**.
- From the **Mesh** contextual toolbar (Mac and Linux), **Boundary** menu (△), choose **Free Quad**.
- Right-click a **Mesh** node and choose **Free Quad**. For 3D models, this is selected from the **More Operations>** menu.

> The quadrilateral mesh generator does not strictly create only quadrilateral elements. In places of the geometry where it judges it as necessary, it can also create triangular elements.

Then enter the properties for the quadrilateral meshing operation using the following sections:

BOUNDARIES (3D) / DOMAINS (2D)

Define the boundaries (3D) or domains (2D) where you want to create an unstructured quad mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured quad mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create an unstructured quad mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create an unstructured quad mesh. Choose **Manual** in the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the x-scale, y-scale, and z-scale in 3D to positive real numbers. If any of the scale factors are not equal to one, the software scales the geometry in the x, y, and z directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic. They are also useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

Free Tetrahedral

Add a **Free Tetrahedral** node (▲) to create an unstructured tetrahedral mesh. If no selection is specified, this feature creates a mesh on the remaining domains, boundaries, edges and points. You can control the number, size, and distribution of elements by using Size and Distribution subnodes.

To create an unstructured tetrahedral mesh for a domain selection:

- In the **Graphics** window, select the domains. On the **Mesh** ribbon toolbar (Windows) or from the **Mesh** contextual toolbar (Mac and Linux), click the **Free Tetrahedral** (▲) button.

  Then choose the menu item corresponding to the desired predefined element size, for example, Normal. The software creates the resulting tetrahedral mesh by adding and building a Free Tetrahedral node, using the selected domains, with a Size node, using the selected predefined element size, added as a subnode. Alternatively, you can click the button associated with the menu button. Then COMSOL uses the last selected menu item (or Free Tetrahedral (Normal)), as indicated by the tooltip. If you use this menu button with an empty selection the software meshes the remaining, unmeshed geometry.
- Right-click a **Mesh** node and choose **Free Tetrahedral**.

Then define the properties for the tetrahedral meshing operation using the following sections:

DOMAIN SELECTION

Define the domains where you want to create an unstructured tetrahedral mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured tetrahedral mesh for remaining, unmeshed domains.

- Choose **Entire geometry** to create an unstructured tetrahedral mesh in the entire geometry.
- Choose **Domain** to specify the domains for which you want to create an unstructured tetrahedral mesh. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains** to select all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the *x*-scale, *y*-scale, and *z*-scale to positive real numbers. If any of the scale factors are not equal to one the software scales the geometry in the *x*, *y*, and *z* directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic, and they are useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

For a tutorial about free meshing and mesh sizing, see Free Tetrahedral Meshing of a Piston Geometry, model library path: **COMSOL_Multiphysics/Meshing_Tutorials/piston_mesh**.

Free Triangular

Add a **Free Triangular** node (△) to create an unstructured triangular mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using Size and Distribution nodes.

To create an unstructured triangular mesh:

- On the **Mesh** ribbon toolbar (Windows) from the **Generators>Boundary** (△) menu, choose **Free Triangular**.
- From the **Mesh** contextual toolbar (Mac and Linux), **Boundary** menu (△), choose **Free Triangular**.
- Right-click a **Mesh** node and choose **Free Triangular**. For 3D models, this is selected from the **More Operations>** menu.

Then enter the properties for the triangular meshing operation using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundaries (3D) or domains (2D) where you want to create an unstructured triangular mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured triangular mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create an unstructured triangular mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create an unstructured triangular mesh. Choose **Manual** in the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the *x*-scale, *y*-scale, and *z*-scale in 3D to positive real numbers. If any of the scale factors are not equal to one the software scales the geometry in the *x*, *y*, and *z* directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic, and they are useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

TRIANGULATION

Here you can specify the triangulation method used when creating the triangular mesh. Select **Automatic** (default) to let the software use the best suited method, select **Delaunay** to use a method based on a Delaunay algorithm, or select **Advancing front** to use a method based on an advancing front algorithm.

Mapped

Add a **Mapped** node (▦) to create a structured quadrilateral mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using Size (only the Maximum element size parameter is used) and Distribution subnodes.

To create a mapped quadrilateral mesh for each domain, the mapped mesher maps a regular grid defined on a logical unit square onto each domain. The mapping method is based on transfinite interpolation. The settings in the Size and Distribution nodes used by a Mapped node determine the density of the logical meshes. For the mapping technique to work, the opposite sides of each logical unit square must be discretized by the same number of edge elements.

By default the relationship between the four sides of the logical unit square and the boundaries around a domain is based on a criterion related to the sharpest angle between boundaries. If you want to control this relationship, right-click the **Mapped** node to add an Edge Groups subnode.

2D Mapped Mesh Geometry

For the 2D mapped meshing technique to work properly, the geometry must be reasonably regular. The following conditions must be satisfied:

- Each domain must be bounded by at least four boundary segments.
- Each domain must be bounded by only one connected boundary component (that is, no holes are allowed).
- The domains must not contain isolated or embedded vertices or boundary segments.
- The shape of each domain must not differ significantly from a rectangle.

For a geometry model that does not initially meet these criteria, it is usually possible to modify it so that a mapped mesh is generated, for example, by splitting it into simpler domains.

To create a mapped quadrilateral mesh:

- On the **Mesh** ribbon toolbar (Windows) from the **Generators>Boundary** (△) menu, choose **Mapped**.
- From the **Mesh** contextual toolbar (Mac and Linux), **Boundary** menu (△), choose **Mapped**.
- Right-click a **Mesh** node and choose **Mapped**. For 3D models, this is selected from the **More Operations>** menu.

Then enter the properties for the mapped meshing operation using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundaries (3D) or domains (2D) where you want to create a mapped quad mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify mapped quad mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create a mapped quad mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create a mesh. Choose **Manual** from the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

ADVANCED SETTINGS

In 3D, you can choose between two different interpolation methods in the **Interpolation method** list. This specifies how the mapped meshing operation determines the positions of the interior mesh points. If you select **Transfinite in 2D** the positions of the interior mesh points are determined by transfinite interpolation in the 2D parameter space of the corresponding surface and if you select **Transfinite in 3D** transfinite interpolation is done in 3D to determine these positions. Select **Auto** to let the mapped meshing operation determine a suitable interpolation method automatically.

Select the **Adjust evenly distributed edge mesh** check box to allow the mapped mesher to automatically adjust the mesh on edges that are not already meshed and where no explicit distribution is applied.

For an example of a 2D mapped mesh, see Tubular Reactor: model library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

One-Point Map

Use a **One-Point Map** node ( ) to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node.

To add this subnode, right-click the Copy Face or Copy Domain node and select **One-Point Map** from the context menu. Then enter the properties using the following sections:

POINT SELECTION

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

- Activate the **Point on source** list and select the point that you want to define as source point in the **Graphics** window.
- Activate the **Point on destination** list and select the point that you want to define as destination point in the **Graphics** window.

Reference

Use a **Reference** node ( ) to refer to another meshing sequence. Building a **Reference** node runs the operation nodes of the referenced sequence. If you have a Scale node preceding a **Reference** node, or as a subnode to a **Reference** node, you can create a finer or coarser version of the mesh generated by the referenced sequence.

To refer to another meshing sequence, right-click a **Mesh** node and select **Reference** in the **More Operations** submenu. Then use the following section to specify the sequence to reference:

REFERENCE

Select the meshing sequence to reference.

It is possible to *expand* a reference (that is, replacing the reference with a copy of the referred sequence). If the reference node has a **Scale** subnode, the attribute nodes in the expanded sequence are scaled accordingly. In some cases, such scaling of attributes cannot be done explicitly, and additional scale nodes are created instead.

To expand a reference, right-click a reference node and select **Expand** ( ).

Refine

Use a **Refine** ( ) node to refine a mesh by splitting elements.

To refine a mesh:

- On the **Mesh** ribbon toolbar (Windows) from the **Operations>Modify** ( ) menu, choose **Elements>Refine** ( ).
- From the **Mesh** contextual toolbar (Mac and Linux), from the **Modify>Elements** menu, choose **Refine** ( ).
- Right-click a 2D or 3D **Mesh** node and from the **More Operations** submenu select **Refine**.

Then use the following sections to specify the parts of the mesh to refine and the method used to refine the elements:

DOMAIN SELECTION

Define the domains where you want to refine the mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to refine the entire mesh.
- Choose **Domain** to specify the domains for which you want to refine the mesh. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

REFINE OPTIONS

Refinement Method

From the **Refinement method** list, select **Regular refinement** to use the regular refinement method or select **Split longest side** to use the refinement method that splits the longest side when refining the mesh. The regular refinement method divides each element into four triangular elements of the same shape in 2D or eight tetrahedral elements of the same shape in 3D. The longest refinement method bisects the longest edge of each element. For 2D geometries COMSOL Multiphysics defaults to the regular refinement method, and in 3D the refinement method that splits the longest side is the default. In 1D, COMSOL always uses regular refinement, where it divides each element into two elements of the same shape.

Number of Refinements

Enter the number of consecutive mesh refinements in the **Number of refinements** field (the default is one refinement).

> It is only possible to refine domains meshed with simplex elements (that is, segments in 1D, triangles in 2D, and tetrahedra in 3D). To refine the mesh in other domains, you must convert the mesh into simplex elements using the Convert node.

REFINE ELEMENTS IN BOX

Check **Specify bounding box** to refine the mesh only within a box. If you refine the mesh only on certain domains, the mesh is refined only in the intersection between the box and the domains.

Specify the box either by entering the coordinates of the lower left corner and upper-right corner of the box or click **Draw box** to interactively specify the box (only available in 2D).

Scale

Use a **Scale** node ( ) to scale the properties of the Size, Distribution, and Boundary Layer Properties nodes. It is possible to add Scale nodes both as global nodes and as local nodes to Reference nodes. A Scale node that exists as a global node affects the size of the mesh elements generated by the subsequent operation nodes. A Scale node that exists as a subnode to a Reference node affects the size of the mesh elements generated by the Reference node only.

If two or more Scale nodes exist on the same selection, the resulting scale factor on that selection is the product of the given scale factors.

To add this as a global node, right-click a **Mesh** node and select **Scale**. To add this as a subnode right-click a Reference node and select **Scale**. See also Global vs. Local Attribute Nodes.

GEOMETRIC SCOPE

In this section you define the geometric entities where you want to specify a scale. Choose the level of the geometry from the **Geometric entity level** list.

- Choose **Entire geometry** to specify the scale for the entire geometry.
- Choose **Domain** to specify the domains for the scale specification. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Boundary** to specify the boundaries for the scale specification. Choose **Manual** from the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.
- Choose **Edge** to specify the edges for the scale specification. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges. This option is only available in 3D.
- Choose **Point** to specify the points for the scale specification. Choose **Manual** from the **Selection** list to select the points in the **Graphics** window or choose **All points** to select all points. This option is only available in 2D and 3D.

SCALE

Specify the scale factor in the **Element size scale** field.

A scale factor less than 1 gives smaller (more) elements; a scale greater than 1 gives larger (fewer) elements.

It is not possible to use a coarser mesh size setting on a geometric entity adjacent to a higher dimensional entity with a finer mesh size setting; the finer setting on a geometric entity overrides the coarser setting on its boundary. Therefore a scale factor larger than 1 might have no effect if the dimensional level of the selection is less than the space dimension.

Size

Use a **Size** node ( ) to specify the size of mesh elements. It is possible to add Size nodes both as global nodes and as local nodes. If there are several Size nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last Size node in the sequence.

A meshing sequence corresponding to a nonempty geometry contains a Size node at the first position in the sequence. This Size node, referred to as the *default Size node*, is defined for the entire geometry and cannot be removed. To override the mesh size settings defined by this Size node, add another Size node to the sequence.

To add this node as a global node, right-click a **Mesh** node and select **Size**. To add this as a subnode, right-click a Mesh Operations node and select **Size**. Also see Global vs. Local Attribute Nodes.

GEOMETRIC SCOPE

This section is not available for the default Size node.

In this section you define the geometric entities where you want to specify a size. Choose the level of the geometry from the **Geometric entity level** list (only available in 3D):

- Choose **Entire geometry** to specify the size for the entire geometry.
- Choose **Domain** to specify the domains for the size specification. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Boundary** to specify the boundaries for the size specification. Choose **Manual** from the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.
- Choose **Edge** to specify the edges for the size specification. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges. This option is only available in 3D.
- Choose **Point** to specify the points for the size specification. Choose **Manual** from the **Selection** list to select the points in the **Graphics** window or choose **All points** to select all points. This option is only available in 2D and 3D.

ELEMENT SIZE

In the **Calibrate for** list, select the physics for which the element size is calibrated. The options available in some cases require module-dependent physics interfaces as indicated. In all cases, the default values for the **Element Size Parameters** are adjusted for the type of problem being solved.

- For any module, **General physics** and **Fluid dynamics** is an option.

- If you have the Plasma Module, and are using one of the plasma interfaces, choose **Plasma** from the **Calibrate for** list.
- If you have the Semiconductor Module, and are using the Semiconductor interface, choose **Semiconductor** from the **Calibrate for** list.

The default element size is **Predefined** and set as **Normal**. See Mesh Element Quality and Size and Predefined Mesh Element Sizes for details about the options. This automatically determines the parameters that you can otherwise customize under **Element Size Parameters**.

Select **Custom** if you want to change the value for any parameters in the **Element Size Parameters** section.

ELEMENT SIZE PARAMETERS

This section is available when **Custom** is chosen as the **Element Size**. Specify all element size parameters using numerical values or user-defined parameters.

Except for the default Size node, if you select a **Custom** element size above, the check boxes are automatically added next to each field. Click to select a check box to activate, and edit, the corresponding parameter.

The following parameters control the mesh element size (the parameters where you define a size use the geometry's length unit). Defaults vary based on whether it is a default node or not. Edit the default values as required for the following fields:

- **Maximum element size**: To limit the allowed element size. For example, if you want to limit the maximum element size to a fraction of the wavelength to make sure that the wave propagation is fully resolved. By using a parametric sweep to vary the maximum element size you can solve the model using meshes with different mesh density to study how it affects the solution.
- **Minimum element size**: To specify the minimum allowed element size. You can use this value to, for example, prevent the generation of many elements around small curved parts of the geometry. This is not available in 1D.
- **Maximum element growth rate**: To determine the maximum rate at which the element size can grow from a region with small elements to a region with larger elements. The value must be greater or equal to one. For example, with a maximum element growth rate of 1.5, the element size can grow by at most 50% (approximately) from one element to another.
- **Curvature factor**. To determine the size of boundary elements compared to the curvature of the geometric boundary (it is the ratio between the element size and the radius of curvature). The curvature radius multiplied by the curvature factor, which must be a positive scalar, gives the maximum allowed element size along the boundary. A smaller curvature factor gives a finer mesh along curved boundaries. This is not available in 1D.
- **Resolution of narrow regions**: To control the number of layers of elements that are created in narrow regions (approximately). The value must be a nonnegative scalar. A higher value gives a finer mesh in narrow regions. If the value of this parameter is less than one, the mesh generator might create elements that are anisotropic in size in narrow regions.

It is not possible to use a coarser mesh size setting on a geometric entity adjacent to a higher dimensional entity with a finer mesh size setting; the finer setting on a geometric entity overrides the coarser setting on its boundary. A warning is given when coarser settings are overridden.

Meshing Operations and Attributes

Swept

The **Swept** node ( ) creates a swept mesh on a domain in 3D by sweeping the mesh from the source face along the domain to an opposite destination face. The source and destination can consist of several connected faces.

You can control the number, size, and distribution of elements using the **Size** and **Distribution** subnodes. The **Swept** node only reads properties from Size nodes defined on the entire geometry or on the domain level and Distribution nodes defined on the domain level.

In domain selection mode this button works in the same way as the **Free Tetrahedral** button. In boundary selection mode the software creates a swept mesh on the remaining domains using the selected boundaries as source faces.

To create a swept mesh:

- On the **Mesh** ribbon toolbar (Windows) or from the **Mesh** contextual toolbar (Mac and Linux), click the **Swept** ( ) button.
- Right-click a 3D **Mesh** node and select **Swept**.

- About Swept Meshes
- Structured Meshes

DOMAIN SELECTION

Specify the domains where you want a swept mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify swept mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to specify swept mesh for the entire geometry.
- Choose **Domain** to specify the domains for which you want a swept mesh. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

SOURCE FACES

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

To specify the source faces directly, activate the **Source Faces** list and select the faces defining the source of the sweep operation in the **Graphics** window.

DESTINATION FACES

To specify the destination faces directly, activate the **Destination Faces** list and select the faces defining the destination of the sweep operation in the **Graphics** window.

SWEEP METHOD

Face Meshing Method

In the **Face meshing method** list you can specify how the unmeshed source faces, which are meshed automatically by the Swept node, are meshed:

- Select **Quadrilateral (Generate hexahedrons)** to generate a surface mesh with quadrilateral elements. This is the default meshing method, but it does not work for all surfaces.
- Select **Triangular (Generate prisms)** to generate a surface mesh with triangular elements.

Sweeping Path

Use **Swept path calculation** if you want to specify the shape of the sweep path. **Sweep following straight lines** means that all interior mesh points are located on straight lines between the corresponding source and destination points. **Sweep following circular arcs** means that all interior mesh points are located on circular arcs between the corresponding source and destination points. **Sweep using interpolation** means that the positions of the interior mesh points are determined by a general interpolation procedure. The default, **Automatic**, means that the sweeping algorithm automatically tries to determine if the sweep path is straight or circular; otherwise, the general approach is used.

Destination Mesh

Use **Destination mesh generation** if you want to specify the method to be used for transferring the source mesh to the destination. **Use a rigid transformation** means that the destination mesh is created by a rigid transformation of the source mesh, **Morph source onto destination** means that the destination mesh is created from the source mesh by a morphing technique, and **Project source mesh onto destination** means that the destination mesh is created from the source mesh by a projection technique. The default, **Determine suitable method**, means that the algorithm automatically tries to determine a suitable method for creating the destination mesh.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

LINKING FACES

You can choose between two different interpolation methods for the linking faces in the **Interpolation method for linking faces** list. This specifies how the mapped mesher, which is used by the swept mesher for the linking faces, determines the positions of the interior mesh points. For more information on the different option see Mapped.

- Thin-Layer Diffusion: model library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**
- See Deformation of a Feeder Clamp: model library path **COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp**
- Joule Heating of a Microactuator: model library path **COMSOL_Multiphysics/Multiphysics/thermal_actuator_jh**

Two-Point Map

Use a **Two-Point Map** node ( ) to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node.

To add a Two-Point Map node as a subnode to a **Copy Face** or a **Copy Domain** node, right-click the node and select **Two-Point Map** from its context menu. Then enter the properties using the following sections:

SOURCE POINTS

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

- Activate the **First point on source** list and select the point that you want to define as first source point in the **Graphics** window.
- Activate the **Second point on source** list and select the point that you want to define as second source point in the **Graphics** window.

DESTINATION POINTS

Activate the **First point on destination** list and select the point that you want to define as first destination point in the **Graphics** window.

Activate the **Second point on destination** list and select the point that you want to define as second destination point in the **Graphics** window.

Importing and Exporting Meshes

About Mesh Export, Import, and Operations on Imported Meshes

It can be useful to import meshes already created by external software or, alternatively, to export a mesh generated by COMSOL Multiphysics into other software. Importing an externally generated mesh can be helpful when a mesh is already saved in a file and recreating the geometry would be difficult and time consuming.

The partitioning of the mesh into domains, boundaries, edges, and points is essential to set up the physics of each Component node. The available operations for imported meshes deliver some basic, but flexible, functionality. It is also possible to export a mesh for use in another software or for external manipulation of the mesh data.

Exporting a Mesh

You can export a mesh to a COMSOL Multiphysics file (`.mphbin` for a binary file format or `.mphtxt` for a text file format) or to a NASTRAN file (`.nas`, `.bdf`, `.nastran`, or `.dat`).

A 3D mesh can also be exported to an STL file.

To open the **Export Mesh** page, right-click the **Mesh** node and select **Export to File** ( ) from the menu.

Select a file type among the available formats in the **File type** list and enter a filename including the path in the **Filename** field (or click **Browse** to specify the filename).

Click the **Export** button to export a mesh to the specified file. A confirmation message appears in the **Messages** window.

If you export to an STL file, COMSOL exports boundary elements only.

If you export to a COMSOL Multiphysics file or to a NASTRAN file, you can specify the dimensions of the elements to export and also choose to include or exclude the geometric entity information. It is also possible to set the field format and to specify if linear or quadratic mesh elements are exported to a NASTRAN file.

Importing Meshes

You can import a mesh from a COMSOL Multiphysics native file or from another meshing sequence. In 3D you can also import meshes from NASTRAN, STL, or VRML files. In 2D you can also import 2D meshes from NASTRAN (the third coordinate must then be the same for all mesh points).

Importing a mesh clears the geometry defined in the corresponding geometry sequence. It is not possible to use an imported mesh together with a user-defined geometry.

When a mesh is imported into COMSOL, the Import node automatically determines a partitioning of the mesh into domains, boundaries, edges, and points. If the automatically performed partitioning does not match the requirements, you can modify the face partitioning by manually adjusting the corresponding parameters.

To import additional meshes, add another Import node. Then COMSOL adds the elements and points of the newly imported mesh to the existing mesh.

Meshes from different Import nodes form an assembly.

Using Several Meshing Sequences of Imported Mesh Type

You can define several meshing sequences for the same geometry (see Adding, Editing, and Building Meshing Sequences). If the geometry sequence is empty (a necessary condition for the **Imported mesh** sequence type), the first **Mesh** node under the **Meshes** node defines a topology and is referred to as the *master sequence*. All the other **Mesh** nodes should define a geometry topologically similar to the one defined by the master sequence. Two geometries are considered to be similar if they have the same number of geometric entities and their points have the same coordinates.

When you build a non-master sequence, COMSOL Multiphysics first builds the master sequence. If the build of the master sequence fails or if the geometries defined by these two sequences are not similar, an error occurs.

If you want to use the geometric multigrid solver, several meshing sequences must be added first.

Import and Multigrid

IMPORTING EXTERNALLY GENERATED MESH DATA

It is possible to import externally generated mesh data using a COMSOL mesh file. The file format contains a section with mesh points coordinates, followed by sections with mesh element information, divided into separate subsections for each mesh element type (see Mesh in the chapter *The COMSOL File Formats* in the *COMSOL API Reference Manual*).

Importing Incomplete Mesh Data

A COMSOL mesh contains elements for all space dimension levels. For example, a tetrahedral mesh consists of domain (tetrahedrons), boundary (triangles), edge, and vertex elements. Furthermore, each element has an index to the geometric entity it belongs to. If a mesh file is incomplete—for example, if it only contains tetrahedrons—the Import operation automatically generates the missing element data. To illustrate this behavior, import the file `mesh_example_1.mphtxt` from

```
\models\COMSOL_Multiphysics\Meshing_Tutorials\
```

This file contains domain elements only with geometric entity information dividing the mesh into two domains. Now, export the imported mesh to a file using the default settings. Then, compare the resulting file (see `mesh_example_4.mphtxt`) with the file `mesh_example_1.mphtxt` and note that the exported file contains complete mesh information; that is, it contains domain elements, boundary elements, edge elements, vertex elements, and geometric entity information.

Transferring Domain Information

If you have an externally generated mesh with a predefined partitioning of the elements, you can transfer this partitioning to COMSOL by specifying geometric entity information in the `.mphtxt` file. To illustrate this, import the file `mesh_example_2.mphtxt`. This file contains domain elements only, without any geometric entity information. The imported mesh consists of one domain only. Note that the imported mesh from the file `mesh_example_1.mphtxt` consists of two domains according to the given geometric entity information.

Transferring Boundary Information

To transfer boundary partitioning information of an externally generated mesh you need to include boundary elements with the corresponding geometric entity information in the `.mphtxt` file. To illustrate this, import the file *mesh_example_3.mphtxt* with the **Boundary partitioning** option set to **Minimal**. This file contains domain and boundary elements with geometric entity information defining 5 boundaries. Note that the imported mesh also has 5 boundaries. Now import the file `mesh_example_1.mphtxt`, which has no boundary information, using the same import settings. Note that the imported mesh now has 3 boundaries only because the Minimal option generates the minimal possible partitioning that is required by the topological criteria.

Operations on Imported Meshes

The following mesh import operation nodes make it possible to define the partitioning of an imported mesh into domains, boundaries, edges, and points, with respect to the physics settings of the Component.

TABLE 8-4: OPERATIONS ON AN IMPORTED MESHES

ICON	NAME AND LINK	USE AND DESCRIPTION
	Ball	To split geometric entities in an imported mesh by an element set defined by a ball.
	Box	To split geometric entities in an imported mesh by an element set defined by a box.
	Create Vertex	To create an additional vertex in an imported mesh.
	Cylinder	To split geometric entities in an imported mesh by an element set defined by a cylinder.
	Delete Entities	To delete geometric entities from an imported mesh.
	Join Entities	To join adjacent geometric entities in an imported mesh.
	Logical Expression	To split geometric entities in an imported mesh by specifying a logical expression.

Using Operations on an Imported Mesh

The following example shows how you can use the mesh import operations to control the partitioning of an imported mesh.

IMPORTED MESH

The following overview is based on using an imported mesh from the **feeder_clamp** model, found in the COMSOL Multiphysics model library and shown in Figure 8-4.

Deformation of a Feeder Clamp: model library path **COMSOL_Multiphysics>Structural_Mechanics>feeder_clamp**

Figure 8-4: The Deformation of a Feeder clamp model showing an imported mesh, which is divided into 5 domains, 69 faces, 174 edges, and 114 points.

JOIN ENTITIES

To form a single domain, use a **Join Entities** ( ) node, which operates on the domain level (that is, add **All domains** to the selection). As a result, you can obtain a mesh for the model with a single domain.

DELETE ENTITIES

To remove all edges, use a **Delete Entities** ( ) node, which operates on the edge level (that is, add **All edges** to the selection). As a result, you can obtain a mesh for the model with no edges or points.

CYLINDER

To define a boundary that defines the contact between the feeder and the clamp, use a **Cylinder** ( ) node that operates on the boundary level (that is, add **All boundaries** to the selection, and use `10.001` as a cylinder radius, `0` and `-20` for top and bottom, `(15, 0, 35)` as position, and **y-axis** as the axis type).

Figure 8-5: Using a Cylinder node to define the contact between the feeder and the clamp.

LOGICAL EXPRESSIONS

To define two boundaries that define screw channels, use a **Logical Expression** ( ) node, which operates on the boundary level (that is, add **All boundaries** to the selection and use `(y+10)^2+(z-55)^2<=4` as the expression).

Figure 8-6: Using a Logical Expression to create boundaries on the two screw channels of the feeder clamp.

BALL

To create a boundary defining one of the washers used for the boundary loads of the model, use a **Ball** ( ) node, which operates on the boundary level (use `(5,-10, 55)` as a ball center and 3.5 as a ball radius). The input boundary selection must be limited; otherwise, the ball operation also splits one of the cylinder boundaries, which was created by the **Logical Expression** node.

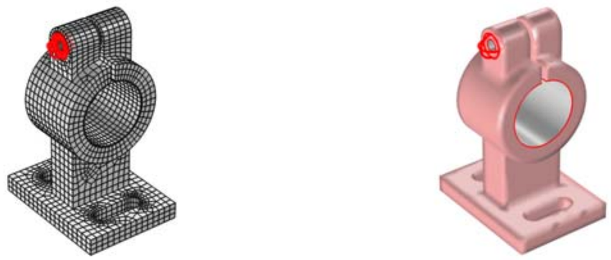


Figure 8-7: Using a Ball node to define one of the washers of the feeder clamp.

By creating a duplicate of the **Ball** node and modifying the ball center (set `x` to `5`) you can create a boundary for the second washer.

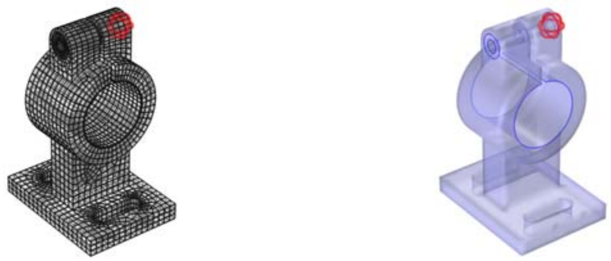


Figure 8-8: Using a Ball node to define a second washer.

BOX

To create the boundaries for the mounting holes, use a **Box** ( ) node, which operates on the boundary level (use `(0 - 30, -30 - 10, 0.1 - 4.9)` as box limits and use the **Some vertex** condition).

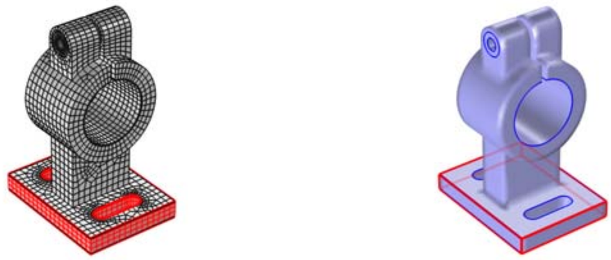


Figure 8-9: Using a Box node to define the mounting holes on the feeder clamp.

CREATE VERTEX

Using the **Create Vertex** ( ) node it is possible to add an additional vertex in a specified location (use `(30,-30, 0)` as vertex coordinates as in Figure 8-10).

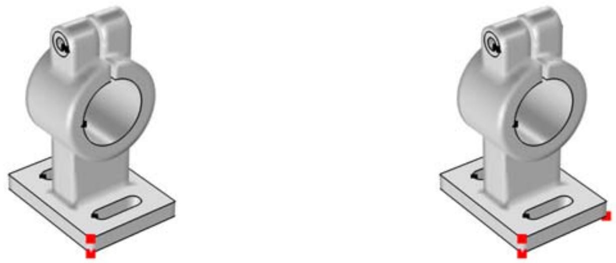


Figure 8-10: Using a Create Vertex node to add vertices at specific locations on the feeder clamp.

Ball

Use a **Ball** node ( ) to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified ball.

To add a Ball node, right-click a 2D or 3D mesh node and select **Ball**. Then use the following sections to specify the geometric entities to split, the properties of the ball, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified ball.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

BALL CENTER

Specify the center of the ball in the **x**, **y**, and **z** (only in 3D) fields (SI unit: m).

BALL RADIUS

Specify the radius of the ball in the **Radius** field (SI unit: m). The default radius is 1.

CONDITION

Use the **Include element if ball contains** list to select the condition for which the element is enclosed in the specified ball. Choose **All vertices** to consider an element to be enclosed in the specified ball if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Box

Use a **Box** node ( ) to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified box.

To add a Box node, right-click a 2D or 3D mesh node and select **Box**. Then use the following sections to specify the geometric entities to split, the properties of the box, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified box.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

BOX LIMITS

Specify the limits of the box in the **x minimum**, **x maximum**, **y minimum**, **y maximum**, **z minimum** (3D only), and **z maximum** (3D only) fields.

CONDITION

Use the **Include element if box contains** list to select the condition for which an element is enclosed in the specified box. Choose **All vertices** to consider an element to be enclosed in the specified box if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Create Vertex

Use a **Create Vertex** node ( ) to create an additional vertex in the closest mesh point to a specified position of an imported mesh.

To add a Create Vertex node, right-click a 2D or 3D mesh node and select **Create Vertex**. Then use the following section to specify the position of the new vertex:

CREATE VERTEX CLOSEST TO POINT

Use the **x**, **y**, and **z** (3D only) fields to specify the position of the vertex. The vertex appears in the mesh point closest to the specified position.

Cylinder

Use a **Cylinder** node ( ) to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified cylinder.

To add a Cylinder node, right-click a 3D mesh node and select **Cylinder**. Then use the following sections to specify the geometric entities to split, the properties of the cylinder, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified cylinder.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

SIZE AND SHAPE

Specify the radius of the cylinder in the field **r** and the positions of the upper and lower boundary circles in the **Top distance** and **Bottom distance** fields, respectively.

POSITION

Specify the position of cylinder in the **x**, **y**, and **z** fields.

AXIS

Use **Axis type** to set the direction of the cylinder axis. Choose **x-axis**, **y-axis**, or **z-axis** to let the cylinder axis coincide with one of the coordinate axes. It is also possible to customize the cylinder axis by choosing **Cartesian** or **Spherical** and using **x**, **y**, and **z**, or **theta** and **phi**, respectively.

CONDITION

Use the **Include element if cylinder contains** list to select the condition for an element to be enclosed in the specified cylinder. Choose **All vertices** to consider an element to be enclosed in the specified cylinder if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Delete Entities

Use a **Delete Entities** node ( ) to delete geometric entities from an imported mesh.

To add a Delete Entities node, right-click a 2D or 3D mesh node and select **Delete Entities**. Then use the following sections to specify the geometric entities to delete:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to delete. You choose the geometric entity level from the **Geometric entity level** list: Choose **Domain**, **Boundary**, **Edge**, or **Point** to specify the domains, boundaries, edges, or vertices, respectively, that you want to delete. Use **All domains**, **All boundaries**, **All edges**, or **All points** to select all entities of the specified dimension.

ADJACENT ENTITIES

Select the **Delete adjacent lower dimensional entities** check box to also delete the adjacent entities of lower dimensions.

Finalize

The **Finalize** node ends a meshing sequence of imported type. It performs an associativity update for geometric entity numbers. You cannot delete, disable, or move the **Finalize** node. The software automatically builds all nodes in a meshing sequence, including the **Finalize** node, if you select a node in Model Builder outside the meshing sequence.

Import

Use an **Import** node ( ) to import a mesh from file or other meshing sequence. It is only possible to import a mesh if the geometry sequence is empty. If the sequence already contains a mesh, the imported mesh is added to the existing mesh, forming an assembly.

To import a mesh, right-click a **Mesh** node and select **Import**. Then enter the properties for the import using the following section:

IMPORT

In the **Mesh source** list choose the type of data to import—**Any importable file**, **Meshing sequence**, and **COMSOL Multiphysics file** are always available. In addition, you can choose **STL/VRML file** in 3D and **NASTRAN file** in 2D and 3D.

For file import, specify the file name in the **Filename** field or click the **Browse** button. For import from another mesh in the model, select the meshing sequence from the **Source** list below. To import a mesh, click the **Import** button ( ). If you have changed some property, the software automatically re-imports the mesh when you click a build button.

Properties for NASTRAN Import

You can import 3D meshes (and planar 2D meshes) in the NASTRAN bulk data format, the most common format for exchanging 3D meshes among programs. This format supports hundreds of NASTRAN entries describing elements, loads, and materials, making it possible to define a complete finite element model. When you import a NASTRAN bulk data file into COMSOL Multiphysics, the software imports mesh and material information only.

To import mesh and material data from a NASTRAN file select **Mesh and materials** in the **Data to import** list. Select **Only mesh** to import the mesh only.

For information on the NASTRAN entries that COMSOL Multiphysics supports, see `Import` in the *COMSOL API Reference Manual* (*Meshing*).

Generally, a NASTRAN bulk data file contains only solid elements. This means that COMSOL enriches the imported mesh data with boundary elements, edge elements, and vertex elements such that a valid mesh object is formed. Each element in the imported mesh object also gets a unique label.

To use element types in the file to determine the domain partitioning of the domain elements, select the **Split on element type** check box (deselected by default). To use material data in the file to determine the domain partitioning of the domain elements, select the **Split on material data** check box (selected by default).To import the elements in the NASTRAN file as linear elements (that is, ignoring node points not in element vertices), select the **Import as linear elements** check box (not selected by default).

Boundary Partitioning Properties

If the partitioning of the boundary elements in the mesh to import into boundaries (faces) is not complete you can use the **Boundary partitioning** list to control the partitioning:

- Select **Automatic** to let the software partition the boundary elements into boundaries (faces) automatically (the default setting).
- Select **Minimal** to make a minimal boundary partitioning. This is useful when you import a mesh from a measured geometry or a NASTRAN mesh with a predefined boundary partitioning. The automatic face partitioning is not desired then.
- Select **Manual** to manually control the partitioning using the following parameters that become available:

If the face partitioning of the boundary elements in the mesh to import is not complete, the boundary elements are automatically partitioned into faces so that you get edges where neighboring triangles' normals make a large angle. To control this algorithm, change **Boundary partitioning** to **Manual**. The value in the **Maximum angle within boundary** field then limits the angle between any two boundary elements in the same boundary (face). The angle between neighboring boundary elements in the same boundary (face) is kept less than the value in the **Maximum boundary neighbor angle** field.

If the **Detect planar boundaries** check box is selected, the mesh import detects (approximately) planar faces (boundaries). A planar face has an area larger than the total area of all faces multiplied by the value in the **Minimum relative area** field. The angle between neighboring boundary elements in the same planar face is kept less than the value in the **Maximum neighbor angle** field.

The following settings are only available for importing 3D meshes:

Removal of Small Boundaries

The mesh import removes small boundaries (faces) with an area that is less the mean area of all faces multiplied by the value in the **Removal of small boundaries** field.

Advanced parameters

If **Manual** is selected from the **Advanced parameters** list, some additional controls for detection of extruded faces and faces with constant curvature become visible. In an extruded face all boundary elements are (approximately) orthogonal to the extruded plane (work plane). The angle between such a triangle's normal and the extruded plane is kept less than the value in the **Maximum angle to extruded plane** field (in degrees). An extruded boundary (face) has an area larger than the total area of all faces multiplied by the value in the **Detect extruded boundaries** field. In a boundary (face) with constant curvature, the relative deviation of the curvature at neighboring triangles is at most the value (in degrees) in the **Maximum curvature deviation in boundary** field. A boundary (face) with constant curvature gets an area larger than the total area of all faces multiplied by the value in the **Detect constant curvature** field.

Join Entities

Use a **Join Entities** node ( ) to join adjacent geometric entities in an imported mesh.

To add a Join Entities node, right-click a 2D or 3D mesh node and select **Join Entities**. Then use the following sections to specify the geometric entities to join:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to join. You choose the geometric entity level from the **Geometric entity level** list: Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to join. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

ADJACENT ENTITIES

Select the **Join adjacent lower dimensional entities** check box to also join the adjacent entities of lower dimensions (boundaries and edges for joined domains, for example).

Logical Expression

Use a **Logical Expression** node ( ) to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements that fulfill the specified logical expression.

To add a Logical Expression node, right-click a 2D or 3D mesh node (that has an imported mesh) and select **Logical Expression**. Then use the sections below to specify the geometric entities to split, the expression, and the split condition.

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified cylinder.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

EXPRESSION

Enter a logical expression using **x**, **y**, or **z** (3D only), Unary, Binary, and List Operators and Their Precedence Rules, and Mathematical and Numerical Constants. For instance, the expression `(x*x+y*y)<1` defines a ball split in 2D and an infinite cylinder split in 3D.

By default, the expression is set to 1 not inducing any split of geometric entities.

CONDITION

Use the **Include element if expression is fulfilled for** list to select the condition for which the logical expression is fulfilled for an element. Choose **All vertices** to make an element satisfy the expression if it is true for all element vertices, or choose **Some vertex** if it is true for at least one element vertex.

Mesh Object

If you open a model created in the 3.5a version of COMSOL Multiphysics a **Mesh Object** node representing the mesh appears in the meshing sequence to handle backward compatibility.

The settings window for the **Mesh Object** node ( ) contains the follow section:

MESH OBJECT

If you save the model as a `.java` file, COMSOL Multiphysics uses the filename specified in the **Filename** field to determine the path to a mesh file, containing the mesh, that appears together with the `.java` file. The software uses this mesh file when you run the resulting `.java` file. By default, the filename has the prefix $FILENAME$. If the filename starts with this prefix, COMSOL puts the mesh file in the same directory as the `.java` file. It is also possible to remove this prefix and specify the full path to the mesh file.

To create a new mesh for a geometry with a **Mesh Object** node in its meshing sequence you first need to delete the **Mesh Object** node.

Meshing Examples

Generating a 3D Swept Mesh

Figure 8-11 shows the 3D Swept mesh for a simple geometry but with a layered structure typical for printed circuit boards or MEMS geometries. In such cases, the swept mesh generation presents an alternative to using a free tetrahedral meshing.

If you have the:

- Acoustics Module, see Vibrations of a Disk Backed by an Air-Filled Cylinder: model library path **Acoustics_Module/Verification_Models/coupled_vibrations_acsh**.
- Batteries & Fuel Cells Module, see Ohmic Losses and Temperature Distribution in a Passive PEM Fuel Cell: model library path **Batteries_and_Fuel_Cells_Module/PEMFC/passive_pem**.
- Electrodeposition Module, see Electrodeposition of an Inductor Coil: model library path **Electrodeposition_Module/Tutorial_Models/inductor_coil**.
- Chemical Reaction Engineering Module, see Steam Reformer: model library path **Chemical_Reaction_Engineering_Module/Heterogeneous_Catalysis/steam_reformer**.

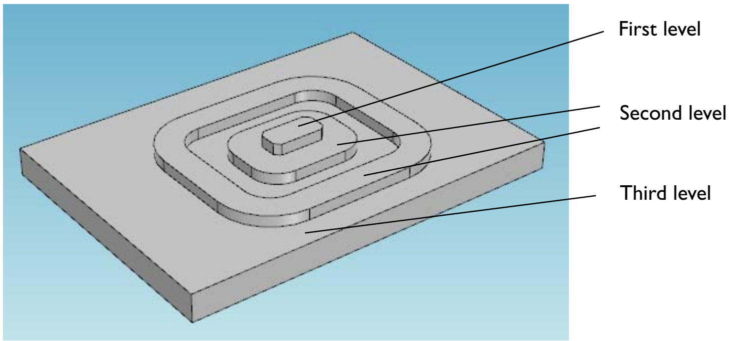
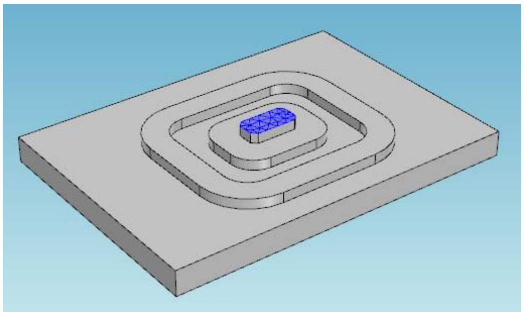


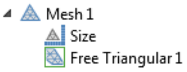
Figure 8-11: An example of the layered geometry used for creating a swept mesh.

1 Add a **Free Triangular** ( ) node from the **Mesh** toolbar, **Boundary** menu (△) (or right-click the **Mesh** node and select it from the **More Operations** menu.)

2 Add the first level boundary to the selection list (see Figure 8-11 for an example of a suitable geometry).

3 Click **Build Selected** ( ). The mesh below displays.





Mesh 1
Size
Free Triangular 1

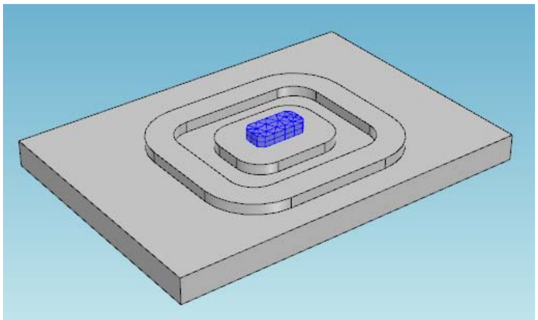
4 Add a **Swept** ( ) node from the **Mesh** toolbar.

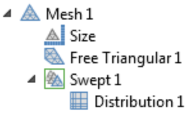
5 Select the domain in the first level.

6 Add a **Distribution** ( ) node to the **Swept 1** node.

7 Enter the **Number of elements** in the field (for example, 2).

8 Click **Build Selected** ( ).

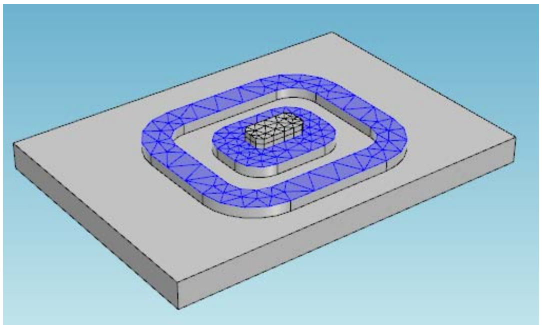


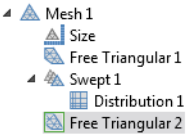


Mesh 1
Size
Free Triangular 1
Swept 1
Distribution 1

9 Add a second **Free Triangular** ( ) node from the **Mesh** toolbar.

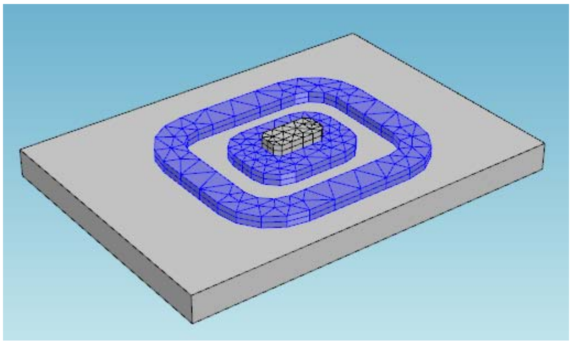
10 Select the boundaries at the second level and click the **Build Selected** button( ).

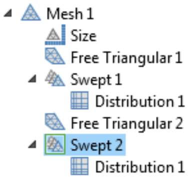




Mesh 1
Size
Free Triangular 1
Swept 1
Distribution 1
Free Triangular 2

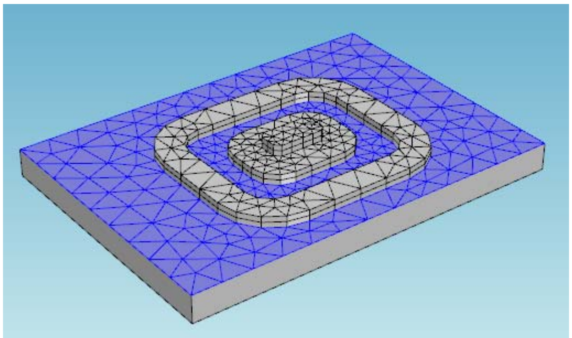
11 Repeat the same swept operations for the first level domains but now for the second level: Add the second **Swept** and **Distribution** nodes.

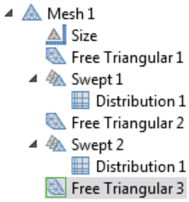




- Mesh 1
  - Size
  - Free Triangular 1
  - Swept 1
    - Distribution 1
  - Free Triangular 2
  - Swept 2
    - Distribution 1

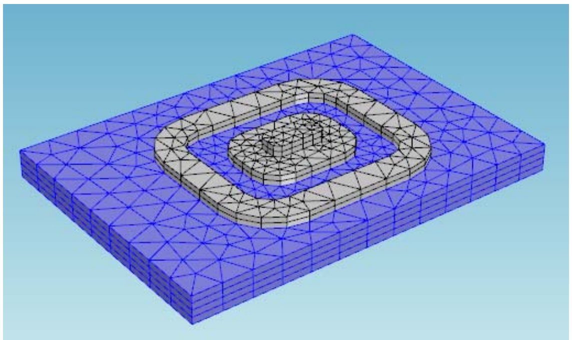
12 Add a third **Free Triangular** mesh operation to Mesh the third level boundaries.

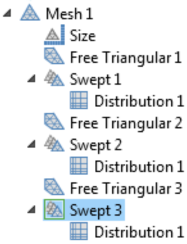




- Mesh 1
  - Size
  - Free Triangular 1
  - Swept 1
    - Distribution 1
  - Free Triangular 2
  - Swept 2
    - Distribution 1
  - Free Triangular 3

13 Mesh the third level domain. Use the **Swept** mesh operation and enter 4 for the **Number of elements** in the corresponding **Distribution** attribute.





- Mesh 1
  - Size
  - Free Triangular 1
  - Swept 1
    - Distribution 1
  - Free Triangular 2
  - Swept 2
    - Distribution 1
  - Free Triangular 3
  - Swept 3
    - Distribution 1

The meshing sequence displayed in the **Model Builder** makes it possible to return to your attribute settings and change mesh sizes and distributions. After making any changes, click the **Build All** button ( ) or press F8 to rebuild the entire meshing sequence.

Using Mesh Control Entities to Control Element Size

Figure 8-12 shows a 2D geometry with two holes and a Bézier Polygon that is intended not to be a part of the model but is included only to control mesh size inside the domain. This example is about Mesh Control Entities and uses a simple geometry.

1 Add a **Mesh Control Edges** ( ) node from the **Geometry** toolbar, **Virtual Operations** menu ( ) (or right-click the **Geometry** node and select it from the **Virtual Operations** submenu).

2 Select the edges of the Bézier Polygon in the **Edges to include** selection.

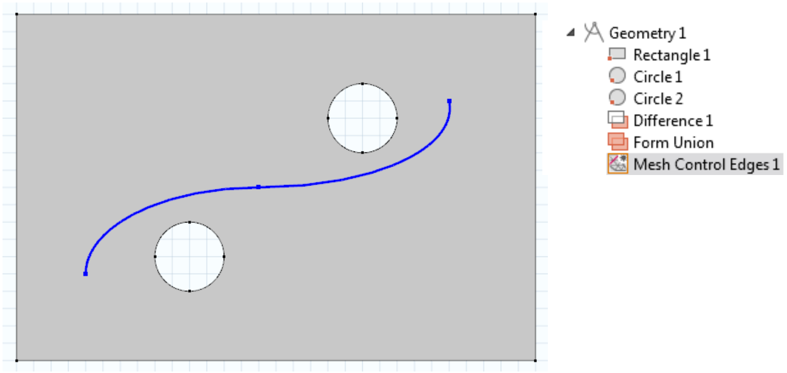


Figure 8-12: A geometry with a Bézier Polygon used to define mesh size inside the domain.

3 Click **Build Selected** ( ). Note that the selected edges are removed.

4 Add a **Free Triangular** ( ) node from the **Mesh** toolbar. Note that the edges removed in the previous step are now visible again.

5 Add a **Size** ( ) node to **Free Triangular 1**.

6 Select Boundary as the Geometric entity level, and select the edges of the Bézier Polygon.

7 Select **Extra fine** as the **Predefined** element size.

8 Click the **Build All** button ( ) or press F8 to build the entire mesh. Note that the edges of the Bézier Polygon are now removed (Figure 8-13) and that the only trace of them is the fine mesh size inside the domain.

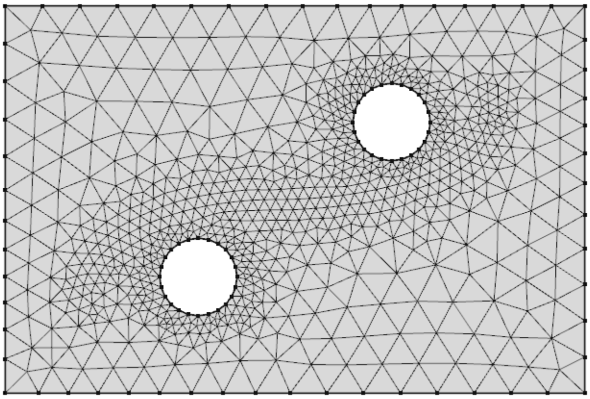


Figure 8-13: Fine mesh inside the domain.

MESH CONTROL FEATURE MODEL EXAMPLES

For an example of the Mesh Control Edges feature:

- If you have the CFD Module, see Turbulent Flow Over a Backward Facing Step: model library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**.
- If you have the Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: model library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**.

For an example of the Mesh Control Faces feature:

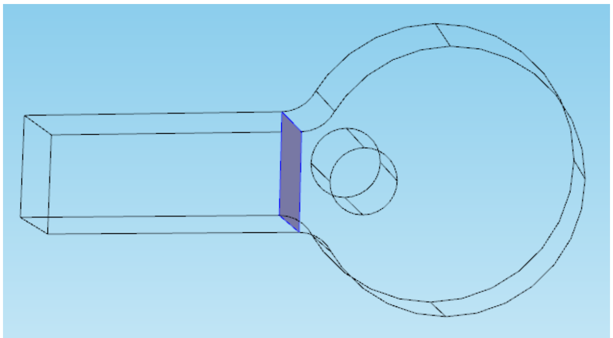
- If you have the Batteries & Fuel Cells Module, see Thermal Modeling of a Cylindrical Li-ion Battery in 3D: model library path **Batteries_and_Fuel_Cells_Module/Batteries/li_battery_thermal_3d**.
- If you have the CFD Module, see Airflow Over an Ahmed Body: model library path **CFD_Module/Single-Phase_Benchmarks/ahmed_body**.

Using Structured and Unstructured Mesh with Boundary Layers

This example demonstrates a geometry where free tetrahedral mesh is used in one domain, and a swept mesh is used in another domain. The domains are separated by a mesh control face, which is automatically removed once the domains on both sides are meshed. Finally boundary layers are added, without the need to respect the (now removed) mesh control face.

1 Add a **Mesh Control Faces** ( ) node from the **Geometry** toolbar, **Virtual Operations** menu ( ) (or right-click the **Geometry** node and select it from the **Virtual Operations** submenu).

2 Select the face separating the domains in the **Faces to include** selection using the scroll wheel for selecting the interior boundary.

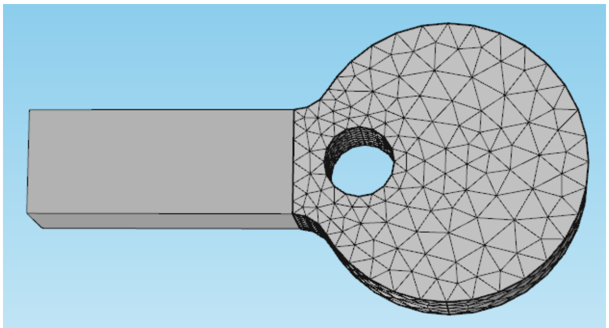


3 Click **Build Selected** ( ). The face is removed. There is now only one domain.

4 Add a **Free Tetrahedral** ( ) node form the **Mesh** toolbar. Note that the face has reappeared, and that there are two domains.

5 Add the cylinder-shaped domain with a hole to the selection list.

6 Click the **Size** ( ) node, and select **Finer** as the **Predefined** element size. Click **Build Mesh** ( ).



7 Add a **Swept** ( ) node from the **Mesh** toolbar.

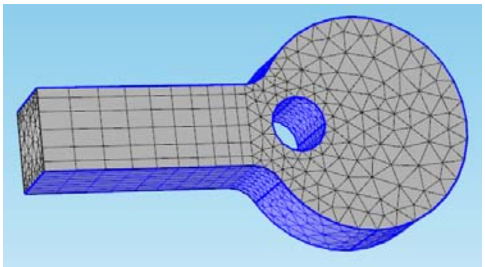
8 Add a **Distribution** ( ) node to **Swept 1** from the **Mesh** toolbar.

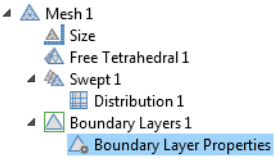
9 Select **Predefined distribution type** and enter 10 in the **Number of elements** field and 3 in the **Element ratio** field. Click **Build Mesh** ( ).

10 Add a **Boundary Layers** ( ) node from the **Mesh** toolbar.

11 In the **Model Builder**, click the **Boundary Layer Properties** ( ) node under **Boundary Layers 1**.

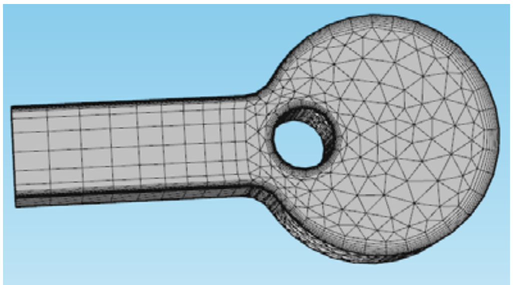
12 Add (for example) the sides of the geometry to the selection list





- Mesh 1
  - Size
  - Free Tetrahedral 1
  - Swept 1
    - Distribution 1
  - Boundary Layers 1
    - Boundary Layer Properties

13 Click the **Build Mesh** button ( ) or press F8 to build the entire mesh. The mesh control face is now removed, and that the boundary layer mesh nodes are not located where the boundary was.



BOUNDARY LAYER MESHING MODEL EXAMPLES

For an example of the Boundary Layer and Boundary Layer Properties features:

- If you have the AC/DC Module, see Iron Sphere in a 20 kHz Magnetic Field: model library path **ACDC_Module/Tutorial_Models/iron_sphere_20khz_bfield**.
- If you have the Acoustics Module, see Acoustic Scattering off an Ellipsoid: model library path **Acoustics_Module/Tutorial_Models/acoustic_scattering**.
- If you have the Batteries & Fuel Cells Module, see Liquid-Cooled Lithium-Ion Battery Pack: model library path **Batteries_and_Fuel_Cells_Module/Batteries/li_battery_pack_3d**.
- If you have the CFD Module, see Turbulent Flow Over a Backward Facing Step: model library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**.
- If you have the Chemical Reaction Engineering Module, see Dissociation in a Tubular Reactor: model library path **Chemical_Reaction_Engineering_Module/Tubular_Reactors/dissociation**.
- If you have the Electrochemistry Module, see Wire Electrode: model library path **Electrochemistry_Module/Electrochemical_Engineering/wire_electrode**.
- If you have the Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: model library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**.
- If you have the Plasma Module, see 3D ICP Reactor, Argon Chemistry: model library path **Plasma_Module/Inductively_Coupled_Plasmas/argon_3d_icp**.

9

Materials

This chapter includes information about how to work with materials in models and describes the material databases included with COMSOL Multiphysics® and the add-on modules.

Materials Overview

About Materials and Material Properties

MATERIALS

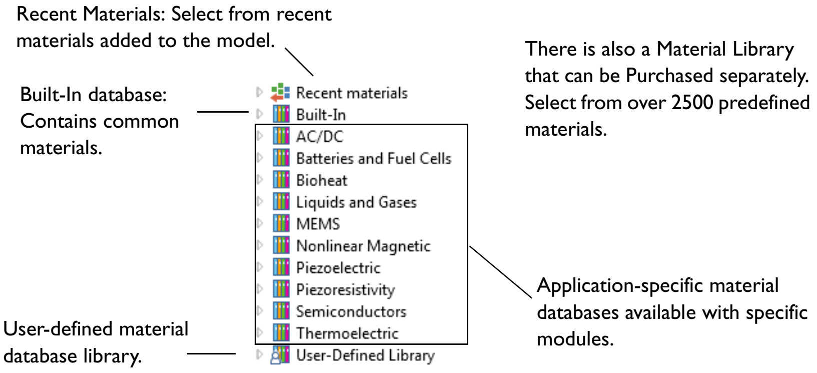
In COMSOL models, you can add one or more *materials*, which are named collections of *material properties*. Each such material is represented by a **Material** node (▦), typically with a name that describes the material, such as **Copper**, under **Component>Materials** in the **Model Builder**. Each material includes a number of physical properties with the values or functions (for temperature-dependent material properties, for example) that describe the material.

MATERIAL PROPERTIES AND PROPERTY GROUPS

The material properties are organized in material *property groups*, which appears as subnodes under the **Material** node in the **Model Builder**:

- The **Basic** property group contains common material properties that can generally be measured and are meaningful without any context.
- User-defined groups may contain a subset of the same quantities.
- Each predefined property group contains one or more material properties that are only meaningful together and in the context of a particular material model.
- The material property values are outputs of the material, which can be constant values or functions of *model inputs* (physical quantities like temperature and pressure) In principle, the physics interfaces first ask a material which inputs it requires to compute its output properties, then asks the material to compute property values given values of the model inputs—for example, thermal conductivity (output) as function of temperature (input).
- Each property group can also define a set of local properties and functions that can be used together with model inputs in output property expressions. This makes it possible to, for example, create generic materials for certain classes of some type of material and use the local properties to parameterize the material.

About the Material Databases



Recent Materials: Select from recent materials added to the model.

Built-In database: Contains common materials.

User-defined material database library.

There is also a Material Library that can be Purchased separately. Select from over 2500 predefined materials.

Application-specific material databases available with specific modules.

Figure 9-1: Use the Material Browser to select predefined materials in all applications.

All COMSOL Multiphysics modules have predefined material data available in collections of materials—material databases—to build models. The most extensive material data is contained in the separately purchased Material Library, but all modules contain commonly used or module-specific collections of materials. For example, the

Built-In database is available to all users but the MEMS database is included with the MEMS Module and Structural Mechanics Module. You can also create custom materials and material libraries by researching and entering material properties.

The **Material Browser** window provides access to all material databases (including the Material Library). The following material databases are available in the Material Browser (some require additional module licenses):

- *Recent Materials*: From the **Recent Materials** folder ( ), select from a list of recently used materials, with the most recent at the top. This folder is available after the first time a material is added to a Component node.
- *Material Library*. An optional add-on database, the **Material Library**, contains data for over 2500 materials and 20,000 property functions.
- *Built-In*. Included with COMSOL Multiphysics, the **Built-In** database contains common solid materials with electrical, structural, and thermal properties.
- *AC/DC*. Included in the AC/DC Module, the **AC/DC** database has electric properties for some magnetic and conductive materials.
- *Batteries and Fuel Cells*. Included in the Batteries & Fuel Cells Module, the **Batteries and Fuel Cells** database includes properties for electrolytes and electrode reactions for certain battery chemistries.
- *Bioheat*: Included in the Heat Transfer Module, the **Bioheat** database includes properties for bone, fat, liver, muscle, prostate, and skin.
- *Liquids and Gases*. Included in the Acoustics Module, CFD Module, Chemical Reaction Engineering Module, Heat Transfer Module, MEMS Module, Pipe Flow Module, and Subsurface Flow Module, the **Liquids and Gases** database includes transport properties and surface tension data for liquid/gas and liquid/liquid interfaces.
- *MEMS*. Included in the MEMS Module and Structural Mechanics Module, the **MEMS** database has properties for MEMS materials—metals, semiconductors, insulators, and polymers.
- *Nonlinear Magnetic*. Included in the AC/DC Module, the **Nonlinear Magnetic** database has properties, such as nonlinear magnetization curves, for a large set of ferromagnetic alloys like various types of steel.
- *Piezoelectric*. Included in the Acoustics Module, MEMS Module, and Structural Mechanics Module, the **Piezoelectric** database has properties for piezoelectric materials.
- *Piezoresistivity*. Included in the MEMS Module, the **Piezoresistivity** database has properties for piezoresistive materials, including p-type and n-type silicon materials.
- *Semiconductors*. Included in the Semiconductor Module, the Semiconductors database contains silicon, gallium arsenide, and germanium materials for use with this module.
- *Thermoelectric*. Included with the Heat Transfer Module and contains bismuth telluride and lead telluride materials for use with the Thermoelectric Effect multiphysics interface.
- *User-Defined Library*. The **User-Defined Library** folder ( ) is where user-defined material databases (libraries) are created. When you have created a new database, it also displays in the **Material Browser**.

The material databases shipped with COMSOL Multiphysics are read-only. This includes the Material Library and any materials shipped with the optional modules.

- Module-Specific Material Databases
- Creating a New Material Library

USING THE MATERIALS IN THE PHYSICS SETTINGS

The physics set-up in a model is determined by a combination of settings in the **Materials** and physics nodes. When the first material is added to a Component node, COMSOL Multiphysics automatically assigns that material to all domains in the geometry (or all boundaries or edges if the Component only contains surfaces or edges). Different geometric entities can have different materials. The following example uses the *heat_sink.mph* model file contained in the Heat Transfer Module and CFD Module model library.

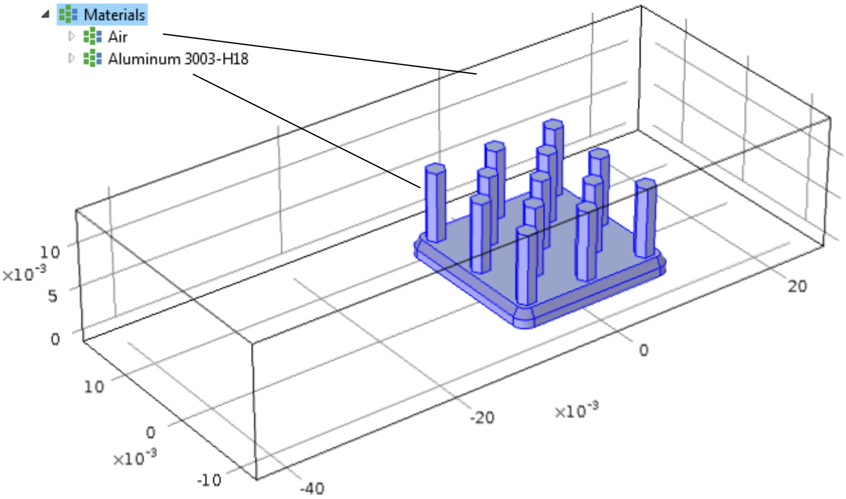


Figure 9-2: Assigning materials to a heat sink model. Air is assigned as the material to the box surrounding the heat sink, and aluminum to the heat sink itself.

If a geometry consists of a heat sink in a container, **Air** can be assigned as the material in the container surrounding the heat sink and **Aluminum** as the heat sink material itself (see Figure 9-2). The **Conjugate Heat Transfer** interface, selected during model set-up, uses a **Fluid** model to simulate non-isothermal flow, with heat transfer by convection and conduction, in the box surrounding the heat sink, and a **Heat Transfer in Solids** model to simulate heat conduction in the heat sink. The **Heat Transfer in Solids 1** settings use the material properties associated to the **Aluminum 3003-H18** materials node, and the **Fluid 1** settings define the flow using the **Air** material properties. The other nodes under **Conjugate Heat Transfer** define the initial and boundary conditions.

All physics properties automatically use the correct material properties from the **Material** nodes when the default **From material** setting is used. This means that one node can be used to define the physics across several domains with different materials; COMSOL then uses the material properties from the different materials to define the physics in each domain.

The Material Settings Window

There are also some physics nodes where you can explicitly select a material from which material properties are retrieved (for example, the **Fluid Properties** node's **Settings** window for two-phase flow modeling). The default setting is then typically to use the **Domain material** on each domain (that is, the materials defined on the same domains as the physics that uses the material data). In addition to the **Domain material**, you can select any other material that is present in the Component, regardless of its selection. The selected material's properties are then applied to all domains in the feature's selection.

EVALUATING AND PLOTTING MATERIAL PROPERTIES

You can access the material properties for evaluation and plotting like other variables in a model using the variable naming conventions and scoping mechanisms:

- To access a material property throughout the model (across several materials) and not just in a specific material, use the special material container `root.material`. For example, `root.material.rho` is the density ρ as defined by the materials in each domain in the geometry. For plotting, you can type the expression `material.rho` to create a plot that shows the density of all materials, for example.

> If you use a temperature-dependent material, each material contribution asks for a special model input. For example, `rho(T)` in a material `mat1` asks for `root.mat1.def.T`, and you need to define this variable (`T`) manually—if the temperature is not available as a dependent variable—to make the density variable work.

- To access a material property from a specific material, you need to know the tags for the material and the property group. Typically, for the first material (Material 1) the tag is `mat1` and most properties reside in the default **Basic** property group with the tag `def`. The variable names appear in the **Variable** column in the table under **Output properties** in the settings window for the property group; for example, `Cp` for the heat capacity at constant pressure. The syntax for referencing the heat capacity at constant pressure in Material 1 is then `mat1.def.Cp`. Some properties are anisotropic tensors, and each of the components can be accessed, such as `mat1.def.k11`, `mat1.def.k12`, and so on, for the thermal conductivity. The numbers 1, 2, and 3 denote the first, second, and third direction, respectively, in the active coordinate system. In the general case, you can define a 3-by-3 tensor, for example, k_{ij} in the order k_{11}, k_{21}, k_{31}, k_{12}, k_{22}, k_{32}, k_{13}, k_{23}, and k_{33}. For material properties that are functions, call these with input arguments such as `mat1.def.rho(pA,T)` where `pA` and `T` are numerical values or variables representing the absolute pressure and the temperature, respectively. Functions can be plotted directly from the function nodes' settings window by first specifying suitable ranges for the input arguments.
- Many physics interfaces also define variables for the material properties that they use. For example, `solid.rho` is the density in the Solid Mechanics interface and is equal to the density in a material when it is used in the domains where the Solid Mechanics interface is active. If you define the density in the Solid Mechanics interface using another value, `solid.rho` represents that value and not the density of the material. If you use the density from the material everywhere in the model, `solid.rho` and `material.rho` are identical.

Working with Materials

The Material Browser Window

The **Material Browser** window ( ) contains a number of databases with a broad collection of elastic, solid mechanics, electromagnetic, fluid, chemical, thermal, piezoelectric, and piezoresistive properties of materials. Use the Material Browser to find predefined materials and add them to the Model Builder, or create a custom material library.

Win

To open the **Material Browser**, from the **Home** ribbon:

- Select **More Windows>Material Browser** .
- From the **Materials** group, click **Browse Materials** .

Mac

Linux

To open the **Material Browser**:

- On the **Model Toolbar** click **Browse Materials** .
- Select **Windows>Material Browser** .

The **Material Browser** is similar to The Add Material Window but it includes detailed property information about each material. From this window you can also create a new material library and import a material library. See Adding Materials to a Component for information about adding materials to your model's components (geometries). Click **Done** ( ) to close the **Material Browser** and add the materials in the **Added to model** list to the model. Click **Cancel** ( ), press Escape, or click in the main toolbar to exit the **Material Browser** without adding any materials.

You can browse all the available material databases or search for specific materials. There is also a **Recent Materials** folder where you find the most recently used materials. **Search** a specific material by name (or, for the Material Library product, by UNS number or DIN number).

When browsing the material databases, in particular the **Material Library**, some materials include additional information—UNS number, DIN number, and composition.

As in Figure 9-3 the following information is included in the window to the right of the material tree. Navigate in the material tree and click a material to display the information.

Material availability is based on the type of COMSOL Multiphysics license. For example, if you have the MEMS Module, you have the **Built-In**, **Liquids and Gases**, **MEMS**, and **Piezoelectric** material libraries.

PROPERTIES

While browsing the databases, predefined material properties for the selected material are listed in a table under **Property** together with their **Expression**, **Unit**, and the **Property group** to which the material property belongs. If **Property group** is empty, the material property is a **Basic** property.

Under **Property reference**, for the materials in the Material Library product, reference information about a material's properties appears when you click a property above.

INPUTS

For some materials, predefined function inputs are listed under **Input** in a table together with the **Input**, **Variable**, and **Unit**. Inputs appear for material properties defined using functions that require the input. Typical inputs are temperature and pressure, for temperature- and pressure-dependent material properties, respectively.

CREATE A NEW MATERIAL LIBRARY OR IMPORT A MATERIAL LIBRARY

Click the **New Material Library** button ( ) to open the **New Material Library** dialog box. You can also right-click a material and choose **Add to New Library** ( ) to create a new material library and add that material to the new library. Go to Creating a New Material Library.

Click the **Import Material Library** button ( ) to open the **Choose Material Library** dialog box. Go to Importing a Material Library.

Figure 9-3: The Material Browser details a material's properties after selection. In this example, the properties of Air, selected from the Built-In library, are listed to the right of the Material Browser folders.

The Add Material Window

The **Add Material** window is similar to The Material Browser Window. It has the same materials available but does not include the detailed properties about each material. This window is a quick way to add materials to models.

To open the **Add Material** window:

- From the **Home** ribbon, click **Add Material** ( ) (Windows).
- From the **Model Toolbar**, click **Add Material** ( ) (Mac and Linux).
- Right-click the **Materials** node ( ) and select **Add Material** ( ).

As in Figure 9-4 You can browse all the available material databases or search for specific materials. There is also a **Recent Materials** folder where you find the most recently used materials. **Search** a specific material by name (or, for the Material Library product, by UNS number or DIN number).

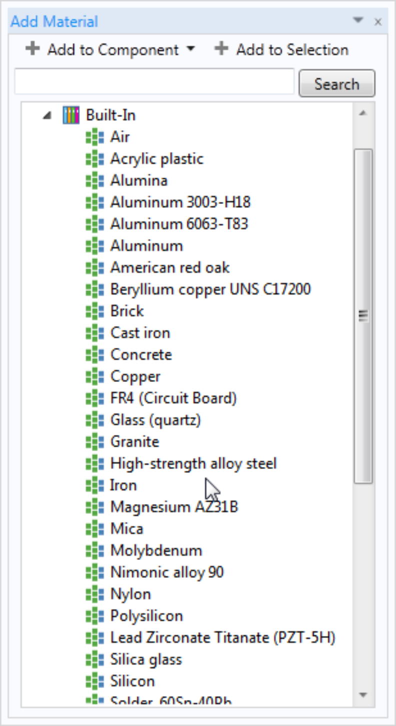


Figure 9-4: The Add Material window. In this example, Air is selected from the Built-In library and can be added to the Material node in the Model Builder.

ADDING MATERIALS TO A COMPONENT

You can add materials to Component nodes using either the **Add Material** or **Material Browser** windows. In either window, use the **Search** field to find materials by name, UNS number, or DIN number. Or click any of the folders and subfolders to locate and add a specific material. For example, from the left column, click **Built-In** and in the right column, click **Air**.

Using the Add Material Window

1 Open The Add Material Window.

2 In the **Add Material** window, choose a material by phase (liquid, vapor, gas, or solid) and orientation/variation, when available.

3 Click the **Add to Selection** or **Add Material to Component** buttons, or right-click the material and choose the same options from the context menu. If there is more than one Component in the model tree, add the material to the applicable geometry.

- If **Add to Selection** is chosen, the material is added to the geometric entity chosen in the Graphics window and a new node is added to the Model Builder. This is a method called preselection and once the second node is added, the first node displays **(overridden)** in the selection list.
- If **Add Material to Component** is selected, the material is added to the **Model Builder** and it becomes the active material in the domains (or other geometric entities) where it is selected. Right-click the **Material** node to rename it, for example, using the name of the material it represents.

Using the Material Browser Window

1 Open The Material Browser Window.

2. In the **Material Browser**, choose options from the **Phase** and **Orientation/variation** lists, when available (only included for some materials in the Material Library product). In this window you can review the material **Properties**, **Functions**, and **Input** sections.
3. Click the **Add to Component** button ( ) under the list of materials to add the selected material to the current model component. Alternatively, click the **Add To** button ( ) to add the material to any available model component or to an existing or new user-defined material library. You can also right-click the selected material node to add that material to a model component or user-defined material library. Materials that you have selected to add to any of the model components appear in the **Added to model** list.
4. Click **Done** (☑) to add the materials to the model tree in the **Model Builder** and close the **Material Browser**. If it is the first material in that model component, the material in the **Model Builder** becomes the default material; otherwise the material is initially not used anywhere but becomes the active material in the domains (or other geometric entities) that you pick to add to that material's selection list.

Materials

Use the nodes under **Materials** ( ) to add predefined or user-defined materials, to specify specific material properties using model inputs, functions, values, and expressions as needed, or to create a custom material library.

MATERIAL OVERVIEW

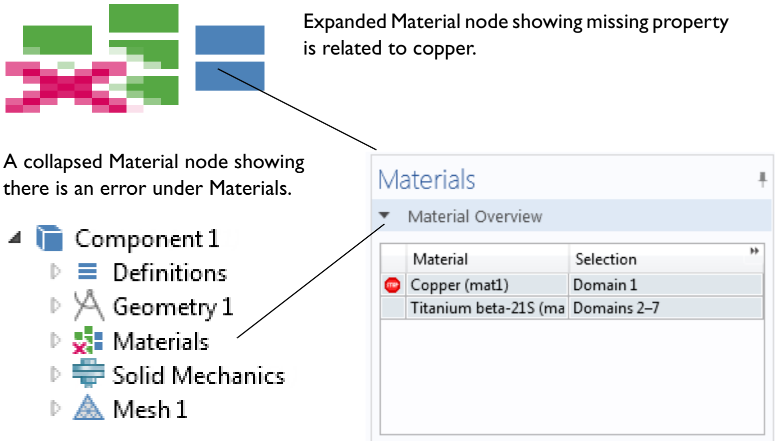
This section provides an overview of the materials in the Component node and where they are used.

The **Material** column lists the current materials in the Component using the materials' node labels from the model tree according to the settings defined in Viewing Node Names, Identifiers, Types, and Tags.

The **Selection** column lists the geometric entities selected for the material (the domains, boundaries, or edges where the material is defined).

ERRORS RELATING TO THE MATERIAL NODES

If a material property in a physics interface takes its value from a material and no material is defined for the same geometric selection, a stop sign (🛑) displays in the leftmost column, and the **Material** column contains **Entities needing a material**. The **Selection** column contains the geometric entities where a material definition is missing. The Material nodes in also indicate when there is a material error. For example, if some property is deleted but needed in a part of the geometry, then the icon indicates where the error is located (see Figure 9-5).



Expanded Material node showing missing property is related to copper.

A collapsed Material node showing there is an error under Materials.

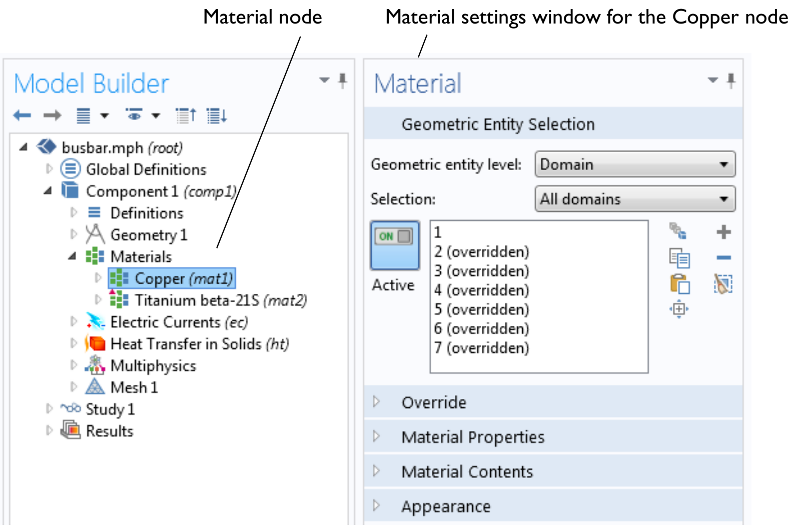
	Material	Selection
🛑	Copper (mat1)	Domain 1
	Titanium beta-21S (ma	Domains 2–7

Figure 9-5: An example of a Material node error.

The Material Settings Window

The **Material** settings window ( ) summarizes the predefined or user-defined material properties for a material. This is where you can add or change material properties to fit your model and assign the material to all types of geometric entities: domains (most common), boundaries, edges (3D models only), or points.

After adding a material, click the material node (for example, **Material 1** or **Copper**) in the **Model Builder**. The **Material** settings window opens.



Geometric entity level: Domain

Selection: All domains

1
2 (overridden)
3 (overridden)
4 (overridden)
5 (overridden)
6 (overridden)
7 (overridden)

Figure 9-6: Click the Copper node to open the Material settings window for the node.

GEOMETRIC ENTITY SELECTION

Assign the material to some or all entities on a specific **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point**—on the geometry in the **Graphics** window (the geometry in the model).

By default, the first material in the Component is active in all domains (or all boundaries or edges if the Component only contains surfaces or edges). By assigning other materials to some or all domains, the first material is overridden and remains active only in domains where no other material, added below it in the **Materials** branch, is active.

If the Component contains features on different geometric entity levels, such as solid mechanics in domains coupled to beams on edges, and the features use the same material, you need to add two Material nodes with the same material, one defined in the domains, and the other defined on the edges.

OVERRIDE

This section shows if the material, in some or all parts of the geometry where it is active, is overridden by another material added underneath it in the Materials branch, or if it overrides another material above it.

The **Overridden by** list shows the names of the materials that override this material. The **Selection** list in the **Geometric Entity List** section displays **(overridden)** for the geometric entities where this material is overridden.

The **Overrides** list shows the names of the materials that this material overrides.

MATERIAL PROPERTIES

You can add material properties to the Component if they are not already included. To do so, browse the available material property categories (**Basic Properties**, **Acoustics**, and so on), and select a material property or a collection of material properties in one of the property groups or material models that appear under the main level of material property categories. Right-click the material property or property group and select **Add to Material**, or click the **Add to material** button (➕) to add the material property or group of properties to the material.

> **!** Review the properties listed in the **Material Contents** table before adding new material properties.

For example, under **Acoustics>Viscous Model** select **Bulk viscosity (muB)** and right-click to **Add to Material** or click the **Add** button (➕). If you add a material model like the **Viscous Model** with more than one property, all its material properties are added to the **Material Contents**. In this example, a **Viscous model** node is added to the **Model Builder**, and its associated properties are added to the **Material Contents** table.

> 💡 To delete a property group, right-click the property group node (in the **Model Builder**) and select **Delete** (☰✕). The **Basic** property group cannot be deleted.

A Note about Adding Basic Material Properties

Material properties can be added to the **Basic** group or to any **User-Defined Property Group** from two locations—the **Material** and **Property Group** settings windows.

- When material properties are added from the **Basic** node's or a user-defined group node's **Property Group** settings window, they are listed under **Output Properties and Model Inputs** in that settings window.
- When material properties are added from the **Material** settings window, the available material properties are listed under **Material Properties** and are added to the list under **Material Contents** with the property group listed. The list under **Material Contents** also contains material properties added from a subnode with a **Property Group** settings window.

Material Type

The **Material type** setting decides how materials behave and how material properties are interpreted when the mesh is deformed. Select **Solid** for materials whose properties change as functions of material strain, material orientation, and other variables evaluated in a material reference configuration (material frame). Select **Non-solid** for materials whose properties are defined only as functions of the current local state at each point in the spatial frame and for which no unique material reference configuration can be defined.

Simply put, **Solid** materials associate material properties with specific pieces of the material, and the properties follow the material as it moves around. In particular, a solid material may be inherently anisotropic, meaning that its axes rotate together with the material. The **Non-solid** choice, in contrast, applies typically to liquids and gases whose properties are associated with fixed points in space and insensitive to local rotation of the material. Such materials are inherently isotropic when studied in isolation, but may exhibit anisotropy induced by external fields. In practice, this means that any anisotropic tensor properties in a Non-solid material must be functions of some external vector field.

MATERIAL CONTENTS

This section lists all the material properties that are defined for the material or required by the physics in the model. The table lists the **Property**, **Name**, **Value**, and **Unit** for the material property as well as the **Property group** that the material property belongs to. The **Property group** corresponds to the subnodes in the **Model Builder** with the same name. If required, edit the values or expression for the property’s **Value**.

The left column provides visual cues about the status of each property:

- A stop sign (🛑) indicates that an entry in the **Value** column is required. It means that the material property is required by a physics feature in the model but is undefined. When you enter a value in the **Value** column, the material property is added to its property group.
- A warning sign (⚠) indicates that the material property has been added to the material but is still undefined. An entry is only required if the material property is to be used in the model.
- A green check mark (✓) indicates that the property has a **Value** and is currently being used in the physics of the model.
- Properties with no indication in the left column are defined but not currently used by any physics in the model.

APPEARANCE

The settings in this section make it possible to control or change the default appearance of a material in the Graphics window when working in the materials or physics parts of the model tree.

In 3D models, the material is rendered including color and texture when **Scene Light** is active. In 2D models and in 3D models when **Scene Light** is turned off, only a change of color is visible.

The **Family** list provides quick settings approximating the appearance of a number of common materials—**Air**, **Aluminum**, **Brick**, **Concrete**, **Copper**, **Gold**, **Iron**, **Lead**, **Magnesium**, **Plastic**, **Steel**, **Titanium**, and **Water**. Select **Custom** to make further adjustments of the specific settings for colors, texture, reflectance, and so on. The default custom settings are inherited from the material selected last from the **Family** list.

The texture and reflectance properties only take effect when the preference settings for the visualization are optimized for quality. When optimized for performance, the appearance includes color only. To set this, open The Preferences Dialog Box, click **Graphics and Plot Windows** and click to select the **Show material color and texture** check box to display material texture and color.

Specular Color, Diffuse Color, and Ambient Color

For each of these properties, click the **Color** button to assign a **Custom** specular color or select a standard color from the list—**Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White**, or **Yellow**.

The combination of **Specular color**, **Diffuse color**, and **Ambient color** gives a 3D object its overall color:

- **Specular color** is the color of the light of a specular reflection (specular reflection is the type of reflection that is characteristic of light reflected from a shiny surface).
- **Diffuse color** represents the true color of an object; it is perceived as the color of the object itself rather than a reflection of the light. The diffuse color gets darker as the surface points away from the light (shading). As with Ambient color, if there is a texture, this is multiplied by the colors in the texture, otherwise it is as if it has a white texture.
- **Ambient color** is the color of all the light that surrounds an object; it is the color seen when an object is in low light. This color is what the object reflects when illuminated by ambient light rather than direct light. Ambient color creates the effect of having light hit the object equally from all directions. As with Diffuse color, if there is

a texture, this is multiplied by the colors in the texture, otherwise it is as if it has a white texture.

For examples of specular, diffuse, and ambient light, which are related to these definitions, see About the 3D View Light Sources and Attributes.

Noise

The **Noise** check box is selected by default, with the default **Normal vector noise scale** and **Normal vector noise frequency** taken from the material. Enter other values as required, or click to clear the **Noise** check box.

- Noise is a texture that disturbs the normals when calculating lighting on the surface. This causes the surface to look rough and textured.
- Normal vector noise scale is the power of the noise texture, a high value creates a stronger texture of the surface. A value between 0–1 is suitable.
- Normal vector noise frequency is the size of the noise disturbances, a small value creates smaller features on the texture. A value between 0–10 is suitable.

Diffuse and Ambient Color Opacity

The default **Diffuse and ambient color opacity** is 1. Enter a different number as required.

Lighting Model

The default **Lighting model**—**Blinn-Phong** or **Cook-Torrance**—is based on the material. Select **Simple** instead as required.

The different lighting models provide a set of techniques used to calculate the reflection of light from surfaces to create the appropriate shading. For example, a specular highlight is the bright spot of light that appears on shiny objects when illuminated. Specular highlights are important in 3D computer graphics because they provide a strong visual cue for the shape of an object and its location with respect to light sources in the scene.

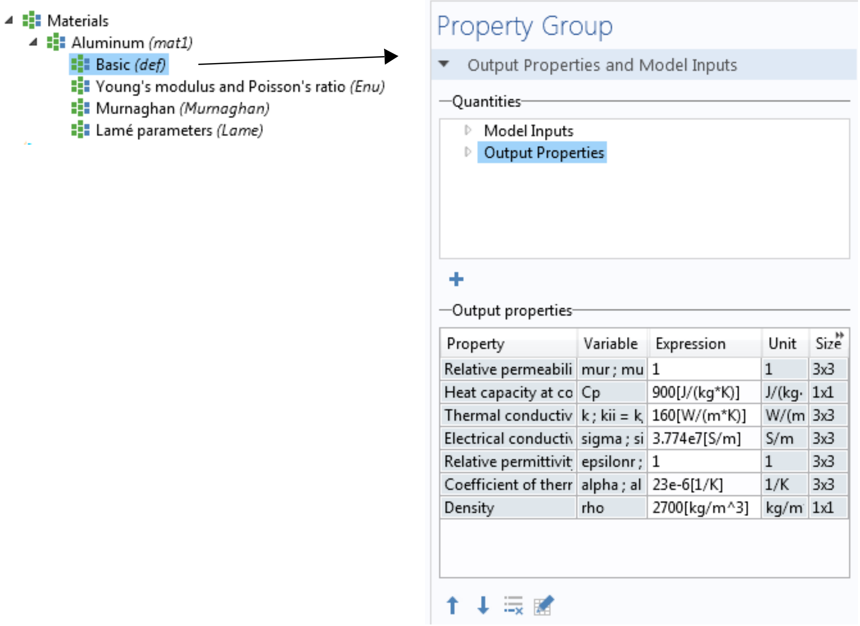
If **Blinn-Phong** is selected, the default **Specular exponent** is 64. Enter another value as required. The specular exponent determines the size of the specular highlight. Typical values for this property range from 1 to 500, with normal objects having values in the range 5 to 20. This model is particularly useful for representing shiny materials.

If **Cook-Torrance** is selected, the default **Reflectance at normal incidence** and **Surface roughness** are taken from the material. Enter other values as required. The Cook-Torrance lighting model accounts for wavelength and color shifting and is a general model for rough surfaces. It is targeted at metals and plastics, although it can also represent many other materials.

- Reflectance at normal incidence is the amount of incoming light (0–1) from the normal direction (of the surface) that is reflected.
- Surface roughness is a value that describes microreflectance on the surface. Higher values create a rougher look of the surface, with fewer highlights. A value from 0–1 is suitable.

Property Groups

The **Property Group** settings window is where output properties and the model inputs are added, local properties are defined, and expressions for material properties are entered in a specific property group such as **Basic**. The property groups are subnodes to a material node. The **Property Group** settings window is displayed when you click the property group node (for example, **Basic**) under the material node (typically with the material's name—**Aluminum**, for example) in the **Model Builder**.



- Materials
  - Aluminum *(mat1)*
    - Basic *(def)*
    - Young's modulus and Poisson's ratio *(Enu)*
    - Murnaghan *(Murnaghan)*
    - Lamé parameters *(Lame)*

Property Group

▼ Output Properties and Model Inputs

Quantities

- Model Inputs
- Output Properties

+

Output properties

Property	Variable	Expression	Unit	Size
Relative permeabili	mur ; mu	1	1	3x3
Heat capacity at co	Cp	900[J/(kg*K)]	J/(kg·	1x1
Thermal conductiv	k ; kii = k	160[W/(m*K)]	W/(m	3x3
Electrical conducti	sigma ; si	3.774e7[S/m]	S/m	3x3
Relative permittivit	epsilonr ;	1	1	3x3
Coefficient of therr	alpha ; al	23e-6[1/K]	1/K	3x3
Density	rho	2700[kg/m^3]	kg/m	1x1

Figure 9-7: An example of a Basic Property Group settings window.

OUTPUT PROPERTIES AND MODEL INPUTS

The predefined material properties in the property group appears in the **Output properties** table. Under **Quantities** you can add additional material properties to the **Output properties** list or add model inputs to the **Model inputs** list.

Output Properties under **Quantities** is only available from the **Basic** material properties and with user-defined property groups.

The model inputs are physical quantities such as temperature that are used as inputs in the expressions that define the output properties (for example, to describe a temperature-dependent physical quantity). For example, adding **Temperature** as a model input with the variable name `T` makes it possible to use an expression for the heat capacity at constant pressure C_p such as `300[J/(kg*K)]*T[1/K]` that works regardless of the name of the actual dependent variable for temperature in the model that uses the temperature-dependent material. Without the model input, the expression above only works with a temperature variable called `T`.

If required, edit the expressions in the **Output properties** list's **Expression** column; edit directly in the table or by clicking the **Edit** button (✎), which opens a dialog box for easier specification of orthotropic and anisotropic material properties (tensors): Select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** when entering the data in the material property's dialog box. In the **Expression** column, use a syntax with curly braces such as `{k11, k21, k31, k12, k22, k32, k13, k23, k33}` to enter anisotropic material properties for a 3-by-3 tensor, for example, k_{ij} in the order k_{11}, k_{21}, k_{31}, k_{12}, k_{22}, k_{32}, k_{13}, k_{23}, and k_{33}. 1, 2, and 3 represent the first, second, and third direction in the active coordinate system.

Use the **Move up** (↑), **Move down** (↓), and **Delete** (☰) buttons to organize the tables as required.

LOCAL PROPERTIES

Here you can enter a user-defined **Property** and its corresponding **Expression** and organize the table as required. These local properties are useful for parameterizing functions that describe material properties, if they contain other inputs than those that are model inputs (such as temperature and pressure). For example, a local property can be a reference value at a certain temperature. Use the **Move up** (↑), **Move down** (↓), and **Delete** (✕) buttons to organize the tables as required.

You can use local properties to parameterize a material (for example, to create a generic “template” material for a particular symmetry class of anisotropic materials). You can then adjust the local property values for each instance of the material.

About Automatic Adding of Property Groups to a Material

Material property groups are automatically added to the material node in the **Model Builder**. You can also add additional predefined property groups or create a **User-Defined Property Group** by right-clicking the Material node. The available properties are collected in property groups according to the physical context.

Each property group has a **Property Group** settings window. When a **Model Builder** node is clicked (for example, **Basic**), the **Property Group** settings window displays specific information about that property group. The physical properties for all property groups are summarized in a **Material Contents** table on the **Material** settings window.

Material Properties Reference

The material properties for the predefined materials are accessible from most physics interfaces. Using this information, either create a material property group or define a completely new material.

In the **Basic>Property Group** window, you can add **Output Properties** under **Quantities** subsection. You can also add **Model Inputs** to, for example, create a temperature- dependent material property.

Model Inputs

The following model inputs (which are scalar or vector-field physical quantities that appear as inputs in, for example, a temperature-dependent material property) can be added to models from the **Property Group** window (vector fields have three components enclosed by curly braces).

TABLE 9-1: MODEL INPUTS

MODEL INPUT	NAME/VARIABLE
Absolute Pressure	pA
Concentration	c
Current Density	{J1, J2, J3}
Electric Field	{E1, E2, E3}
Frequency	freq
Magnetic Field	{H1, H2, H3}
Magnetic Flux Density	{B1, B2, B3}
Number Density	nd
Strain Reference Temperature	Tempref
Stress Tensor	{F1, F2, F3}
Temperature	T
Velocity Field	{u1, u2, u3}

About the Output Material Properties

Some of these material groups are only used by physics interfaces in the add-on modules and detailed information is in the applicable documentation.

This section describes all available property groups and the material properties that they contain. These material properties can be added to models from two settings windows: the **Material** window and its subnodes’ **Property Group** windows.

The Basic group contains over 25 basic properties for use with all materials.

Materials

BASIC MATERIAL PROPERTIES

These common material properties belong to the **Basic** property group.

- When this information is accessed from the **Basic>Property Group** window, it is listed under **Quantities>Output Properties** and **Variable** is listed in the table.
- When this information is accessed from the **Material** window, it is listed under **Material Properties>Basic Properties** and **Name** is listed in the table under **Material Contents**.

TABLE 9-2: BASIC MATERIAL PROPERTIES

PROPERTY	NAME/VARIABLE	SI UNIT
Absorption Coefficient	kappaR	1/m
Bulk Viscosity	muB	Pa·s
Characteristic Acoustic Impedance	Z	Pa·s/m
Coefficient of Thermal Expansion	alpha	1/K
Compressibility of Fluid	chif	1/Pa
Density	rho	kg/m^3
Diffusion Coefficient	D	m^2/s
Dynamic Viscosity	mu	Pa·s
Electrical Conductivity	sigma	S/m
Electron Mobility	mue	m^2/(Vs)
Heat Capacity at Constant Pressure	Cp	J/(kg·K)
Isotropic Structural Loss Factor	eta s	1
Mass Flux	Mf	kg/(m^2·s)
Mean Molar Mass	Mn	kg/mol
Permeability	kappa	m^2
Poisson's Ratio	nu	1
Porosity	epsilon	1
Ratio of Specific Heats	gamma	1
Relative Permeability	mur	1
Relative Permittivity	epsilonr	1
Resistivity	res	Ω·m
Scattering Coefficient	sigmaS	1/m
Seebeck Coefficient	S	V/K
Speed of Sound	cp	m/s
Storage	S	1/Pa
Surface Emissivity	epsilon rad	1
Thermal Conductivity	k	W/(m·K)
Young's Modulus	E	Pa

⚠ The coefficient of thermal expansion (CTE) and the resistivity temperature coefficient have the SI unit 1/K. COMSOL Multiphysics translates this into the Fahrenheit temperature unit using an offset. This means that you do not get the expected results.

Use caution when a model uses the coefficient of thermal expansion or the resistivity temperature coefficient and the unit system's temperature is not kelvin.

The rest of the material properties are grouped by application area:

- Acoustics Material Properties
- Electrochemistry Material Properties
- Electromagnetic Models
- Gas Models
- Piezoelectric Models
- Piezoresistive Models
- Solid Mechanics Material Properties
- Solid Mechanics Material Properties: Nonlinear Structural Materials Module
- Solid Mechanics Material Properties: Fatigue Module
- Solid Mechanics Material Properties: Geomechanics Material Model
- Semiconductors Material Properties

Acoustics Material Properties

Under Acoustics you find two acoustic material models with their associated material properties: a **Poroacoustics Model**, **Thermoacoustics Model**, and a **Viscous Model**.

These material property groups (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Acoustics Module.

TABLE 9-3: ACOUSTICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
POROACOUSTICS MODEL		
Flow resistivity	Rf	Pa·s/m^2
Thermal characteristic length	Lth	m
Viscous characteristic length	Lv	m
Tortuosity factor	tau	1
THERMOACOUSTICS MODEL		
Bulk viscosity	muB	Pa·s
Density	rho	kg/m^3
Dynamic viscosity	mu	Pa·s
Heat capacity at constant pressure	Cp	J/(kg·K)
Thermal conductivity	k	W/(m·K)
VISCOUS MODEL		
Bulk viscosity	muB	Pa·s

Electrochemistry Material Properties

These material property groups for electrochemistry (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Batteries & Fuel Cells Module, Corrosion Module, or Electrodeposition Module.

TABLE 9-4: ELECTROCHEMISTRY MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELECTRODE POTENTIAL		
Equilibrium potential	Eeq	V
Reference concentration	cEeqref	mol/m^3
Temperature derivative of equilibrium potential	dEeqdT	V/K
ELECTROLYTE CONDUCTIVITY		
Electrolyte conductivity	sigmal	S/m
LINEARIZED RESISTIVITY	This material node defines the electric resistivity (and conductivity) as a linear function of temperature.	
Reference resistivity	rho0	Ω·m
Reference temperature	Tref	K
Resistivity temperature coefficient	alpha	1/K
SPECIES PROPERTIES		
Transport number	transNum	1

Electromagnetic Models

These material property groups for various electromagnetic material models (including their associated physical properties) can be added to models from the **Material** window. These properties require the AC/DC Module, RF Module, or Wave Optics Module.

TABLE 9-5: ELECTROMAGNETIC MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
BH CURVE	This material node is only available with the AC/DC Module.	
Local Properties	normH	-
Magnetic flux density norm	normB	T
DIELECTRIC LOSSES		
Dielectric loss factor	eta_epsilon	-
Relative permittivity (imaginary part)	epsilonBis	1
Relative permittivity (real part)	epsilonPrim	1
E-J CHARACTERISTIC	This material node is only available with the AC/DC Module.	
Electric field norm	normE	V7M
Local Properties	normJ	-
HB CURVE	This material node is only available with the AC/DC Module.	
Local Properties	normB	-
Magnetic field norm	normH	A/m

TABLE 9-5: ELECTROMAGNETIC MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
LINEARIZED RESISTIVITY	This material node defines the electric resistivity (and conductivity) as a linear function of temperature.	
Reference resistivity	rho0	Ω·m
Reference temperature	Tref	K
Resistivity temperature coefficient	alpha	1/K
LOSS TANGENT	This material node assumes zero conductivity.	
Loss tangent	delta	-
Relative permittivity (real part)	epsilonPrim	1
MAGNETIC LOSSES		
Relative permeability (imaginary part)	murBis	-
Relative permeability (real part)	murPrim	-
REFRACTIVE INDEX	This material node assumes a relative permeability of unity and zero conductivity. This material node is only available with the RF Module or the Wave Optics Module.	
Refractive index, imaginary part	ki	-
Refractive index	n	1

Gas Models

This material property group for an ideal gas (including its associated physical properties) can be added to models from the **Material** page.

TABLE 9-6: GAS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
IDEAL GAS		
Heat capacity at constant pressure	Cp	J/(kg·K)
Mean molar mass	Mn	kg/mol
Ratio of specific heats	gamma	1
Specific gas constant	Rs	J/(kg·K)

Piezoelectric Models

These material property groups for piezoelectric materials (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Acoustics Module, MEMS Module, or Structural Mechanics Module.

TABLE 9-7: PIEZOELECTRIC MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
STRAIN-CHARGE FORM		
Compliance matrix	sE	1/Pa
Coupling matrix	dET	C/N
Loss factor for compliance matrix	sE	1
Loss factor for coupling matrix	d	1

TABLE 9-7: PIEZOELECTRIC MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
Loss factor for electrical permittivity	εT	1
Relative permittivity	epsilonrT	1
STRESS-CHARGE FORM		
Coupling matrix	eES	C/m^2
Elasticity matrix	cE	Pa
Loss factor for elasticity matrix	cE	1
Loss factor for coupling matrix	e	1
Loss factor for electrical permittivity	εS	1
Relative permittivity	epsilonrS	1

Piezoresistive Models

These material property groups for piezoresistive materials (including their associated physical properties) can be added to models from the **Material** window. These property groups require the MEMS Module.

TABLE 9-8: GAS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELASTORESISTANCE FORM		
Elastoresistive coupling matrix	ml	Ω·m
PIEZORESISTANCE FORM		
Piezoresistive coupling matrix	Pil	A/m^2

Solid Mechanics Material Properties

These material property groups for material models in solid mechanics (including their associated physical properties) can be added to models from the **Material** window.

TABLE 9-9: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
LINEAR ELASTIC MATERIAL		
ANISOTROPIC		
Elasticity matrix	D	Pa
Loss factor for elasticity matrix D	eta_D	1
ANISOTROPIC, VOIGT NOTATION		
Elasticity matrix, Voigt notation	DV0	Pa
Loss factor for elasticity matrix D, Voigt notation	eta_DVo	1
BULK MODULUS AND SHEAR MODULUS		
Bulk modulus	K	N/m^2
Shear modulus	G	N/m^2
LAMÉ PARAMETERS		
Lamé parameter λ	lambLame	N/m^2
Lamé parameter μ	muLame	N/m^2
ORTHOTROPIC		
Young's modulus	Evector	Pa

TABLE 9-9: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	**NAME/VARIABLE**	**SI UNIT**
Poisson's ratio	nuvector	1
Shear modulus	Gvector	N/m^2
Loss factor for orthotropic Young's modulus	eta_Evector	1
Loss factor for orthotropic shear modulus	eta_Gvector	1
ORTHOTROPIC, VOIGT NOTATION		
Shear modulus, Voigt notation	GvectorVo	N/m^2
Loss factor for orthotropic shear modulus, Voigt notation	eta_GvectorVo	1
PRESSURE-WAVE AND SHEAR-WAVE SPEEDS		
Pressure-wave speed	cp	m/s
Shear-wave speed	cs	m/s
YOUNG'S MODULUS AND POISSON'S RATIO		
Young's modulus	E	Pa
Poisson's ratio	nu	1
YOUNG'S MODULUS AND SHEAR MODULUS		
Young's modulus	E	Pa
Shear modulus	G	N/m^2
LINEAR VISCOELASTIC MATERIAL		
Long-term shear modulus	Gv	N/m^2
Bulk modulus	K	N/m^2
POROELASTIC MATERIAL		
Biot-Willis coefficient	alphaB	1
Porosity	epsilon	1
Permeability	kappa	m^2

- The *Geomechanics Module User's Guide* and Table 9-12
- The *Nonlinear Structural Materials Module User's Guide* and Table 9-10
- The *Fatigue Module User's Guide* and Table 9-11

Solid Mechanics Material Properties: Nonlinear Structural Materials Module

These material property groups for material models in solid mechanics using the Nonlinear Structural Materials Module (including their associated physical properties) can be added to models from the **Material** window.

TABLE 9-10: HYPERELASTIC AND ELASTOPLASTIC MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	**NAME/VARIABLE**	**SI UNIT**
ELASTOPLASTIC MATERIAL		
Hardening function	sigmagh	Pa
Hill's coefficients	Hillcoefficients	$(m^2 \cdot s^4)/kg^2$
Initial tensile and shear yield stresses	ys	N/m^2
Initial yield stress	sigmags	Pa
Isotropic tangent modulus	Et	Pa

TABLE 9-10: HYPERELASTIC AND ELASTOPLASTIC MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
Kinematic tangent modulus	Ek	Pa
HYPERELASTIC MATERIALS		
ARRUDA-BOYCE		
Macroscopic shear modulus	mu0	N/m^2
Number of segments	Nseg	1
BLATZ-KO		
Model parameters	phiBK	1
Model parameters	betaBK	1
Shear modulus	muBK	Pa
GAO		
Model parameters	aG	Pa
Model parameters	nG	1
GENT		
Macroscopic shear modulus	muG	Pa
Model parameters	jmG	1
MOONEY-RIVLIN		
Model parameters	C01, C02, C03, C10, C11, C12, C20, C21, C30	Pa
MURNAGHAN	The Murnaghan node adds five model parameters. The model is based on strain invariants and is typically used in acoustoelasticity.	
Murnaghan third-order elastic moduli	l	Pa
Murnaghan third-order elastic moduli	m	Pa
Murnaghan third-order elastic moduli	n	Pa
Lamé parameter λ	lambLame	Pa
Lamé parameter μ	muLame	Pa
VARGA		
Model parameters	c1VA	Pa
Model parameters	c2VA	Pa
YEOH		
Model parameters	c1YE	Pa
Model parameters	c2YE	Pa
Model parameters	c3YE	Pa

Solid Mechanics Material Properties: Fatigue Module

These material property groups for material models in solid mechanics using the Fatigue Module (including their associated physical properties) can be added to models from the **Material** window.

TABLE 9-11: ELASTOPLASTIC AND FATIGUE BEHAVIOR MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELASTOPLASTIC MATERIAL>RAMBERG-OSGOOD		
Cyclic hardening coefficient	K_ROcyclic	Pa
Cyclic hardening coefficient	n_ROcyclic	1
FATIGUE BEHAVIOR>ENERGY-BASED		
DARVEAUX		
Crack initiation energy coefficient	K1_Darveaux	1
Crack initiation energy exponent	k2_Darveaux	1
Crack propagation energy coefficient	K3_Darveaux	m
Crack propagation energy exponent	k4_Darveaux	1
Reference energy density	Wref_Darveaux	J/m^3
MORROW		
Fatigue energy coefficient	Wf_Morrow	J/m^3
Fatigue energy exponent	m_Morrow	1
FATIGUE BEHAVIOR>STRAIN-BASED		
COFFIN-MANSON		
Fatigue ductility coefficient	epsilonf_CM	1
Fatigue ductility exponent	c_CM	1
Shear fatigue ductility coefficient	gammaf_CM	1
Shear fatigue ductility exponent	cgamma_CM	1
FATEMI-SOCIE		
Normal stress sensitivity coefficient	k_FS	1
WANG-BROWN		
Normal stress sensitivity coefficient	S_WB	1
FATIGUE BEHAVIOR>STRESS-BASED		
BASQUIN		
Fatigue strength coefficient	sigmaf_Basquin	Pa
Fatigue strength exponent	b_Basquin	1
Shear fatigue strength coefficient	tauf_Basquin	Pa
Shear fatigue strength exponent	bgamma_Basquin	1
FINDLEY		
Normal stress sensitivity coefficient	k_Findley	1
Limit factor	f_Findley	Pa
MATAKE		
Normal stress sensitivity coefficient	k_Matake	1
Limit factor	f_Matake	Pa

TABLE 9-11: ELASTOPLASTIC AND FATIGUE BEHAVIOR MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
NORMAL STRESS		
Limit factor	f_NormalStress	Pa

Solid Mechanics Material Properties: Geomechanics Material Model

These material property groups for material models in solid mechanics (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Geomechanics Module.

TABLE 9-12: GEOMECHANICS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
CAM-CLAY MATERIAL MODEL		
Swelling index	kappaSwelling	1
Compression index	lambdaComp	1
Initial void ratio	e0	1
Cam-Clay M parameter	M	1
DRUCKER-PRAGER		
Drucker-Prager alpha coefficient	alphaDrucker	1
Drucker-Prager k coefficient	kDrucker	Pa
HOEK BROWN		
Hoek-Brown m parameter	mHB	1
Hoek-Brown s parameter	sHB	1
Geological strength index	GSI	1
Disturbance factor	Dfactor	1
Intact rock parameter	miHB	1
LADE-DUNCAN		
Lade-Duncan k coefficient	kLade	1
MATSUOKA-NAKAI		
Matsuoka-Nakai mu coefficient	muMatsuoka	1
MOHR-COULOMB		
Cohesion	cohesion	Pa
Angle of internal friction	internalphi	rad
OTTOSEN		
Ottosen a parameter	aOttosen	1
Ottosen b parameter	bOttosen	1
Size factor	k1Ottosen	1
Shape factor	k2Ottosen	1
YIELD STRESS PARAMETERS		
Uniaxial tensile strength	sigmaut	Pa
Uniaxial compressive strength	sigmauc	Pa
Biaxial compressive strength	sigmabc	Pa

Semiconductors Material Properties

These material property groups for all the material models in semiconductors (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Semiconductor Module.

The Property Group, Variable Names, and SI Unit columns are applicable to all materials in the Semiconductor Module. However, the Values and References columns listed in Table 9-13 are specifically for Silicon.

TABLE 9-13: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY (ALL MATERIALS)	NAME/VARIABLE (ALL MATERIALS)	SI UNIT	VALUE FOR SILICON	REFERENCE FOR SILICON
BASIC				
Relative permittivity	epsilonr	1	11.7	1
Thermal conductivity	k	W/(m·K)	131 W/(m·K)	1
Density	rho	kg/m^3	2329 kg/m^3	1
Heat capacity at constant pressure	Cp	J/(kg·K)	700 J/(kg·K)	1
SEMICONDUCTOR MATERIAL				
Band gap	Eg0	V	1.12 V (valid at 300 K)	1
Effective density of states, conduction band	Nc	1/m^3	2.8×10^{19} 1/cm^3 $\times$(T/300 K)$^{3/2}$	1
Effective density of states, valence band	Nv	1/m^3	1.04×10^{19} 1/cm^3 $\times$(T/300 K)$^{3/2}$	1
Electron affinity	chi0	V	4.05 V	1
Electron mobility	mun	m^2/(V·s)	1450 cm^2/(V·s)	1
Hole mobility	mup	m^2/(V·s)	500 cm^2/(V·s)	1
GENERATION-RECOMBINATION>AUGER RECOMBINATION				
Auger recombination factor, electrons	Cn	m^6/s	2.8×10^{-31} cm^6/s (valid at 300 K)	2
Auger recombination factor, holes	Cp	m^6/s	9.9×10^{-32} cm^6/s (valid at 300 K)	2
GENERATION-RECOMBINATION>DIRECT RECOMBINATION				
Direct recombination factor	C	m^3/s	0 m^3/s	N/A
GENERATION-RECOMBINATION>IMPACT IONIZATION				
a factor, electrons, impact ionization	an	1/V	0.426 1/V	3
a factor, holes, impact ionization	ap	1/V	0.243 1/V	3
b factor, electrons, impact ionization	bn	V/m	4.81×10^5 V/cm	3

TABLE 9-13: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY (ALL MATERIALS)	NAME/VARIABLE (ALL MATERIALS)	SI UNIT	VALUE FOR SILICON	REFERENCE FOR SILICON
b factor, holes, impact ionization	bp	V/m	6.53×10^{5} V/cm	3
c factor, electrons, impact ionization	cn	1/KValues	3.05×10^{-4} 1/K	3
c factor, holes, impact ionization	cp	1/K	5.35×10^{-4} 1/K	3
d factor, electrons, impact ionization	dn	1/K	6.86×10^{-4} 1/K	3
d factor, holes, impact ionization	dp	1/K	5.67×10^{-4} 1/K	3
GENERATION-RECOMBINATION>SHOCKLEY-READ-HALL RECOMBINATION				
Electron lifetime, SRH	taun	s	10 μs	4
Hole lifetime, SRH	taup	s	10 μs	4
MOBILITY MODELS>ARORA MOBILITY MODEL				
Electron mobility reference	mun0_ref_arora	$m^2/(V\cdot s)$	1252 $cm^2/(V\cdot s)$	5
Hole mobility reference	mup0_ref_arora	$m^2/(V\cdot s)$	407 $cm^2/(V\cdot s)$	5
Electron mobility reference minimum	mun_min_ref_arora	$m^2/(V\cdot s)$	88 $cm^2/(V\cdot s)$	5
Hole mobility reference minimum	mup_min_ref_arora	$m^2/(V\cdot s)$	53.4 $cm^2/(V\cdot s)$	5
Electron reference impurity concentration	Nn0_ref_arora	$1/m^3$	1.26×10^{17} $1/cm^3$	5
Hole reference impurity concentration	Np0_ref_arora	$1/m^3$	2.35×10^{17} $1/cm^3$	5
Alpha coefficient	alpha0_arora	1	0.88	5
Mobility reference minimum exponent	beta1_arora	1	-0.57	5
Mobility reference exponent	beta2_arora	1	-2.33	5
Impurity concentration reference exponent	beta3_arora	1	2.4	5
Alpha coefficient exponent	beta4_arora	$m^2/(V\cdot s)$	-0.146	5
Reference temperature	Tref_arora	K	300 K	5
MOBILITY MODELS>CAUGHEY-THOMAS MOBILITY MODEL				
Electron alpha coefficient	alphan0_ct	1	1.11	6
Electron alpha exponent	betan1_ct	1	0.66	6

TABLE 9-13: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY (ALL MATERIALS)	NAME/VARIABLE (ALL MATERIALS)	SI UNIT	VALUE FOR SILICON	REFERENCE FOR SILICON
Electron saturation velocity	vn0_ct	m/s	1 x 10^7 cm/s	6
Electron velocity saturation exponent	betan2_ct	1	-0.87	6
Hole alpha coefficient	alphap0_ct	1	1.21	6
Hole alpha exponent	betap1_ct		0.17	6
Hole saturation velocity	vp0_ct	m/s	8.37x 10^6 cm/s	6
Hole velocity saturation exponent	betap2_ct	1	-0.52	6
Reference temperature	Tref_ct	K	300 K	6
MOBILITY MODELS>FLETCHER MOBILITY MODEL				
Fletcher mobility coefficient 1	F1_fl	1/(cm·V·s)	1.04×10^{21} 1/(cm·V·s)	7
Fletcher mobility coefficient 2	F2_fl	1/m^2	7.45×10^{13} 1/cm^2	7
Reference temperature	Tref_fl	K	300 K	7
MOBILITY MODELS>LOMBARDI SURFACE MOBILITY MODEL				
Electron delta coefficient	deltan_ls	V/s	5.82 x 10^{14} V/s	8
Electron mobility reference	mun1_ls	m^2/(V·s)	4.75 x 10^7cm^2/(V·s)	8
Electron mobility reference	mun2_ls	m^2/(V·s)	1.74 x 10^5 cm^2/(V·s)	8
Electron alpha coefficient	alphan_ls	1	0.125	8
Hole delta coefficient	deltap_ls	V/s	2.05 x 10^{14} V/s	8
Hole mobility reference	mup1_ls	m^2/(V·s)	9.93 x 10^7 cm^2/(V·s)	8
Hole mobility reference	mup2_ls	m^2/(V·s)	8.84 x 10^5 cm^2/(V·s)	8
Hole alpha coefficient	alphap_ls	1	0.0317	8
Reference temperature	Tref_ls	K	1 K	8
Electric field reference	Eref_ls	V/m	1 V/cm	8
Doping concentration reference	Nref_ls	1/m^3	1 1/cm^3	8
MOBILITY MODELS>POWER LAW MOBILITY MODEL				
Electron mobility reference	mun0_pl	m^2/(V·s)	1448 cm^2/(V·s)	5

TABLE 9-13: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY (ALL MATERIALS)	NAME/VARIABLE (ALL MATERIALS)	SI UNIT	VALUE FOR SILICON	REFERENCE FOR SILICON
Hole mobility reference	mup0_pl	$m^2/(V\cdot s)$	$473\ cm^2/(V\cdot s)$	5
Electron exponent	alphan_pl	1	2.33	5
Hole exponent	alphap_pl	1	2.23	5
Reference temperature	Tref_pl	K	300 K	5

User-Defined Materials and Libraries

User-defined materials provide the flexibility needed to design your model and experiments using a combination of existing material properties and properties you define yourself. You can also create your own material database (library) to include materials you use often.

You can also modify and extend existing materials that you load from any of the material libraries. When added to the Component node, the material is a copy of the properties and the material from the library, and you can modify that material's properties in the same way as a user-defined material.

Importing a Material Library

1 In the **Material Browser** window's toolbar, click the **Import Material Library** ( ) button. The **Choose Material Library** dialog box opens.

2 Navigate to a material library file on your computer. To the right of **File name**, choose **XML File (.xml)**, to find material libraries stored as XML files, which is the standard format.

3 When you have located the file to import, click **Open**.

An example of an external material library is MatWeb. MatWeb provides a service where you can export technical datasheets from MatWeb's collection in the format for a COMSOL material library. For more information about this service, visit www.matweb.com.

Creating a New Material Library

When you first open the **Material Browser**, an empty **User-Defined Library** is available for you to start creating your own library of materials (see Figure 9-1). These steps describe how to copy this existing library and rename it to one suitable for your purposes.

1 Open The Material Browser Window.

2 Click the **New Material Library** button ( ). Or click the **Add Material** button ( ) and choose **New Library** from the menu.

3 In the **New Material Library** dialog box, navigate to the folder on the computer where the empty **User-Defined Library** database is located. The location of the file varies based on your installation. For example, if the installation is on your hard drive:

- The file path on Windows might be similar to `C:\Users\`*`Your_Name`*`\.comsol\v44\materials`.
- On Linux, the file path is typically `~/.comsol/v44/material`.
- On the Mac, it is typically `<`*`home folder`*`>/Library/Preferences/COMSOL/v44/material` (if missing, click the Finder's **Go** menu and hold down the Option key to show the `Library` folder). You can also search for the file name `User_Defined_Library.mph`.

4 Right-click the `User_Defined_Library.mph` and select **Copy**. Rename the copied file. The new name must include underscores (_) between words, for example `My_Metals.mph`.

5 Click **Save**. The empty database, with a new name, is added to the Material Browser.

ADDING A PREDEFINED MATERIAL TO THE USER-DEFINED LIBRARY

1 Open the **Material Browser** and add any predefined material to a Component node in the **Model Builder**.

2 Right-click the **Material** node and select **Add Material to Library** from the context menu.

3 Right-click to **Rename Selected** material (as required) that has been added to your library. See Figure 9-8.

REMOVING A MATERIAL FROM A USER-DEFINED LIBRARY

Open the **Material Browser**. Locate the material to remove. Right-click the material and select **Remove Selected** ( ).

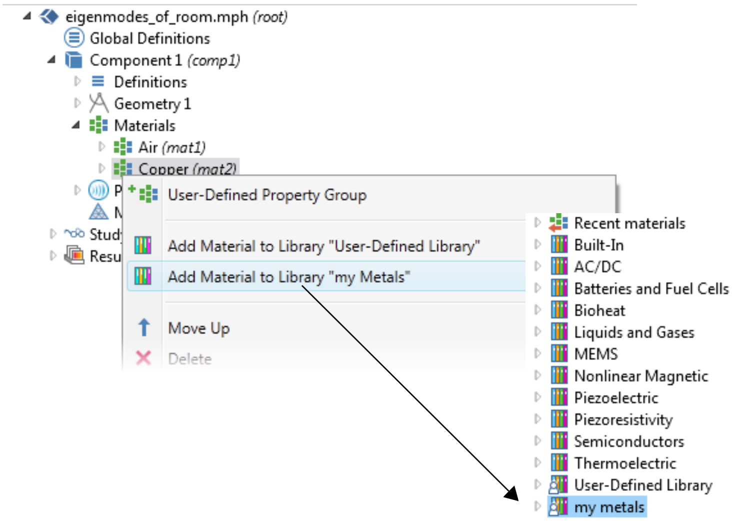


Figure 9-8: Adding a predefined material to a new user-defined library.

Restoring a Deleted User-Defined Library

If the **User-Defined Library** node is deleted in error from the **Material Browser**, you can restore it by following the steps in Creating a New Material Library and then import the file to the Material Browser.

Using Functions in Materials

Functions are useful for describing material properties as, for example, functions of temperature or pressure.

Adding a Function to the Material

Material functions are either automatically added to the **Model Builder** sequence (usually with materials from the model library) or functions can be added based on individual requirements:

1 Add a material to the **Component** node.

2 In the **Model Builder**, right-click a property group node, for example, **Basic**.

3 Select one of the following from the **Functions** list:

- Select **Analytic** to add an analytic function of one or more input arguments.
- Select **Interpolation** to add an interpolation function that can interpolate from structured data (defined on a grid) or unstructured data (defined on a generic point cloud).
- Select **Piecewise** to add a piecewise function that is useful if a material property has different definitions on different intervals.The intervals must not overlap, and there cannot be any holes between intervals.

- Defining an Analytic Function
- Analytic
- Interpolation
- Piecewise

Once a function is created, you can use it for any property in the same property group.

Defining an Analytic Function

Assume that you want to define the density ρ_1 for a material as a function of pressure and temperature: $\rho_1 = \rho_1(p,T)$. You can name the function `rho1(p,T)` and use the expression `p*0.02897/8.314/T` to define the function.

1 Open the **Material Browser**.

2 Add a new material to the Component (or use an existing material where density is not defined, or redefine the current expression for the density).

3 Add a **Density** property to the material.

   a In the **Model Builder**, click the material node.

   b On the **Material** page, click to expand the **Material Properties** section. Under **Basic Properties**, right-click **Density** and **Add to Material**.

   A **Density** property is added to the **Basic** property group.

4 In the **Model Builder**, under the material node, right-click **Basic** and select **Functions>Analytic**. This adds an **Analytic** subnode ( ) under **Basic**.

5 On the **Analytic** settings window, enter `rho1` in **Function Name**.

6 Under **Definition**:

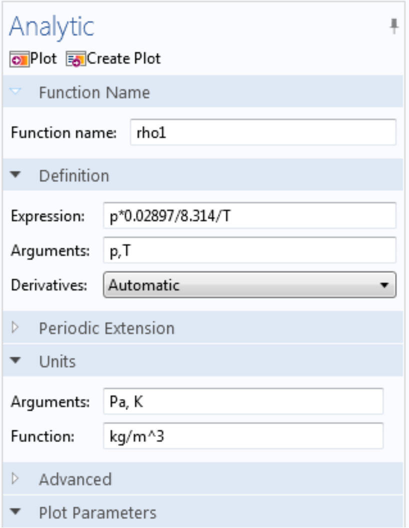
a In the **Expression** field enter `p*0.02897/8.314/T`.

b In the **Arguments** column, enter `p,T`.

7 Under **Units**:

a In the **Arguments** field, enter `Pa, K` as the units for the pressure and the temperature, respectively.

b In the **Function** field: enter `kg/m^3` as the unit for the function’s output (density). The function `rho1` can now be used to define the density in your material.



Analytic

Function Name

Function name: rho1

Definition

Expression: p*0.02897/8.314/T

Arguments: p,T

Derivatives: Automatic

Periodic Extension

Units

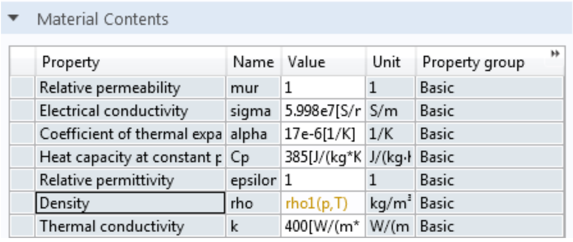
Arguments: Pa, K

Function: kg/m^3

Advanced

Plot Parameters

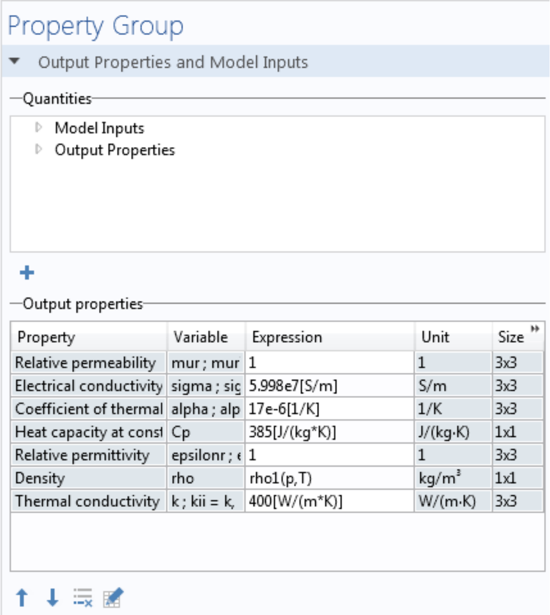
8 Click the material node. On the **Material** settings window, under **Material Contents**, enter `rho1(p,T)` in the **Value** column (in the Density row).



Material Contents

Property	Name	Value	Unit	Property group
Relative permeability	mur	1	1	Basic
Electrical conductivity	sigma	5.998e7[S/r	S/m	Basic
Coefficient of thermal expa	alpha	17e-6[1/K]	1/K	Basic
Heat capacity at constant p	Cp	385[J/(kg*K	J/(kg·l	Basic
Relative permittivity	epsilor	1	1	Basic
Density	rho	rho1(p,T)	kg/m^3	Basic
Thermal conductivity	k	400[W/(m*	W/(m	Basic

Click the **Basic** node to notice that the **Density** analytic function is defined on the **Property Group** settings window under **Output properties**. See Figure 9-9.



Property Group

▼ Output Properties and Model Inputs

Quantities

- ▷ Model Inputs
- ▷ Output Properties

Output properties

Property	Variable	Expression	Unit	Size
Relative permeability	mur ; mur	1	1	3x3
Electrical conductivity	sigma ; sig	5.998e7[S/m]	S/m	3x3
Coefficient of thermal	alpha ; alp	17e-6[1/K]	1/K	3x3
Heat capacity at const	Cp	385[J/(kg*K)]	J/(kg·K)	1x1
Relative permittivity	epsilonr ; ε	1	1	3x3
Density	rho	rho1(p,T)	kg/m^3	1x1
Thermal conductivity	k ; kii = k,	400[W/(m*K)]	W/(m·K)	3x3

Figure 9-9: A density property is defined using an analytic function.

Module-Specific Material Databases

- See About the Material Databases for information about which modules have these material databases.
- For more information about customizing the material's appearance in the Graphics window, see The Material Settings Window.

In this section:

- AC/DC Material Database
- Nonlinear Magnetic Material Database
- Bioheat Material Database
- Liquids and Gases Material Database
- MEMS Material Database
- Piezoelectric Materials Database
- Piezoresistivity Materials Database
- Batteries and Fuel Cells Materials Database
- Semiconductor Materials Database

AC/DC Material Database

The electromagnetic material properties that can be stored in the material databases are:

- Electrical conductivity and resistivity
- Relative permittivity
- Relative permeability
- Nonlinear BH-curves
- Refractive index

The database contains electromagnetic and other material properties for these materials:

MATERIALS
Copper
Soft Iron (without losses)
Soft Iron (with losses)
Quartz
Graphite
Graphite felt
Silicon Carbide

Some properties depend on the magnetic flux density, location, or temperature. The database contains, depending on the material and in addition to the more common material properties, the following properties:

PREDEFINED PROPERTIES
Remnant flux density
Reference temperature

PREDEFINED PROPERTIES
Temperature coefficient
Nonlinear BH-curves
Resistivity at reference temperature

Nonlinear Magnetic Material Database

Nonlinear Magnetic Material Database

The Nonlinear Magnetic materials database is included with the AC/DC Module and has properties, such as nonlinear magnetization curves, for a large set of ferromagnetic alloys like various types of steel.

AC/DC Material Database

MATERIALS
Silicon Steel NGO
35JN200
35PN210, 35PN230, 35PN250, 35PN270, 35PN300, 35PN360, 35PN440
50PN1300, 50PN250, 50PN270, 50PN290, 50PN310, 50PN350, 50PN400, 50PN470, 50PN600, 50PN700, 50PN800
Arnon 5, Arnon 7
M-14, M-22, M-36
Silicon Steel GO
3%
3408, 3411, 3413, 3423
M-6 Cross, M-6 Rolling
Microsil 4 mil
Silectron 12 mil, Silectron 2 mil, Silectron 4 mil cross, Silectron 4 mil rolling, Silectron 6 mil
Trafoperm N3
Metglas
Nano Finemet 50 Hz NoFieldAnnealed
Nano Finemet 50 Hz TFA
Nano FT3M
Nano Nanocrystalline
Nanocrystalline Vitroperm 50 Hz NFA
Nanocrystalline Vitroperm 50 Hz TFA
Nanocrystalline
Vitroperm 400
Vitroperm 50 Hz LFA
Cobalt Steel

MATERIALS
2VPermendur
Cast Cobalt
Cobalt
Supermendur
Vacoflux 17, Vacoflux 50
Vanadium Permendur
Nickel Steel
4750 Cross
4750
Deltamax Oriented
Molypermalloy
Monel Annealed
Monimax, Nonoriented and Monimax, Oriented
Mumetal 77% Ni, Mumetal 80% Ni, Mumetal
Ni 30% Temperature Compensated Alloy
Nickel Annealed
Permalloy Oriented, Permalloy NGO, Permalloy 65% Oriented
Perminvar
Sinimax
Square 50, Square 80
Supermalloy
Superperm 49, Superperm 80
Supersquare 80
Stainless Steel
405 Annealed, 410 Annealed, 416 Annealed, 430 Annealed, 430F Annealed
455 Annealed
Chrome 35% Steel
Chromium Stell
Annealed SUS 403, SUS405 Annealed, Annealed TAF
Low Carbon Steel
1002, 1006, 1008, 1010, 1018, 1020, 1030, 1117, 12L14, 50H470
Cold Rolled Annealed Steel, Cold Rolled Low Carbon Strip Steel
D6ac, M-50
Hot Rolled Steel Strip, Magnet Steel, Magnetite, Soft Iron, Steel Forging Annealed, Tungsten Steel, Vacofer S1 Pure Iron, Pure Iron, Annealed
Low Carbon Iron
Pure Iron
Casting
Cast Iron, Nodular
Cast Iron
Cast Steel
Ductile Iron 3% Si

MATERIALS
Ductile Iron
Gray Iron Gray Iron, As Cast
Ingot Iron Annealed
Ingot Iron
Malleable Iron Malleable Iron, As Cast
Steel Casting, As Cast
Iron Powder
2 Material, 8 Material
SMP 1171, SMP 1172, SMP 1182, SMP 1192, SMP 1220, SMP 1230, SMP 1321
Vetroferrit
Alloy Powder Core
Amoflux
Hiflux 125 mu, Hiflux 14 mu, Hiflux 147 mu, Hiflux 160 mu, Hiflux 26 mu, Hiflux 60 mu
Koolmu 125 mu, Koolmu 26 mu, Koolmu 40 mu, Koolmu 60 mu, Koolmu 75 mu, Koolmu 90 mu
MPP 125 mu, MPP 14 mu, MPP 147 mu, MPP 160 mu, MPP 173 mu, MPP 200mu, MPP 26 mu, MPP 300 mu, MPP 550 mu, MPP 60 mu
Sendust SuperMSS 26, Sendust SuperMSS 60mu, Sendust SuperMSS
Xflux
Ferrite F3000 mu, Ferrite H150000 mu, Ferrite J5000 mu, Ferrite K Material, Ferrite R 2500 mu, Ferrite R 2300 mu
Ferrite

THIRD PARTY NOTICES

The Nonlinear Magnetic Material Database uses data supplied by the BHmag Database, supplied by Precision Wafers Inc. Copyright © 2013 by Precision Wafers Inc.

All data given in this BHmag database is accurate to the best of the provider's knowledge. But it is presented only to assist the user in making informed evaluation and decisions on magnetic materials. No liability is implied, expressed or accepted by the provider for any consequences in usage of the data. The provider does not assume liability to anyone for use of any implicit or explicit illegal use of any copyrighted material. The provider is also not liable for any loss or damage caused by any error or omission in the data, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

Silectron is a registered trademark of ATI Properties, Inc.. Metglas is a registered trademark of Metglas Inc. Super-MSS is a trademark of Micrometals, Inc. Kool Mμ, Xflux and Amoflux are registered trademarks of Spang & Company. Finemet is a registered trademark of Hitachi Metals Ltd.   Trapoferm, Vitroperm, and Vacoflux are registered trademarks and Vacofer is a trademark of Vacuumschmelze GmbH & Co. KG. Monel is a registered trademark of Huntington Alloys Corporation. MuMetal is a registered trademark of Magnetic Shield Corporation. Permalloy is a registered trademark of Marigold Works, LLC. Vetroferrit is a trademark of Von Roll Deutschland GmbH. Microsil, Supersquare 80, Supermendur, Supermalloy, Superperm 80, and Superperm 49 are product names of Magnetic Metals Corporation.

Bioheat Material Database

The Bioheat materials database contains materials used with the Heat Transfer Module Bioheat Transfer interface. Also see References for the Bioheat Materials Database.

MATERIAL	THERMAL CONDUCTIVITY	DENSITY	HEAT CAPACITY	FREQUENCY FACTOR	ACTIVATION ENERGY
Bone	0.32 W/(m·K)	1908 kg/m^3	1313 J/(kg·K)	—	—
Fat	0.21 W/(m·K)	911 kg/m^3	2348 J/(kg·K)	4.43e16	1.3e5
Liver	0.52 W/(m·K)	1079 kg/m^3	3540 J/(kg·K)	7.39e39	2.577e5
Muscle	0.49 W/(m·K)	1090 kg/m^3	3421 J/(kg·K)	—	—
Prostate	0.51 W/(m·K)	1045 kg/m^3	3760 J/(kg·K)	2.984e8	5.064e5
Skin	0.37 W/(m·K)	1109 kg/m^3	3391 J/(kg·K)	4.575e72	4.71e5

Liquids and Gases Material Database

The Liquids and Gases materials database contains thermal and fluid dynamic properties for a set of common liquids and gases. All properties are given as functions of temperature and at atmospheric pressure, except the density, which for gases is also a function of the local pressure. The database also contains surface and interface tensions for a selected set of liquid/gas and liquid/liquid systems. All functions are based on data collected from scientific publications.

TABLE 9-14: LIQUIDS AND GASES MATERIALS

GROUP	MATERIAL	GROUP	MATERIAL
Gases References 1, 2, 7, and 8	Air Nitrogen Oxygen Carbon dioxide Hydrogen Helium Steam Propane Ethanol vapor Diethyl ether vapor Freon12 vapor SiF4	Liquids References 2, 3, 4, 5, 6, 7, 9, and 10	Engine oil Ethanol Diethyl ether Ethylene glycol Gasoline Glycerol Heptane Mercury Toluene Transformer oil Water

MEMS Material Database

The MEMS material database contains several materials commonly used in MEMS applications. The materials are divided into the groups metals, semiconductors, insulators, and polymers.

The basic structure of the library comes from the book *Microsensors, MEMS, and Smart Devices* (Ref. 1). The material properties come from two primary sources: the *CRC Handbook of Chemistry and Physics* (Ref. 2) and *MacMillan's Chemical and Physical Data* (Ref. 3). Some of the mechanical properties in the library are instead more MEMS-specific values from *The MEMS Handbook* (Ref. 4), and most of the semiconductor properties are values from Ref. 5. Most of the PDMS material properties are based on Ref. 7, where Young's modulus is based on Ref. 8.

Ref. 6 provides a valuable resource for cross-checking the insulation material properties.

TABLE 9-15: MEMS MATERIALS

MATERIAL
Metals
Aluminium (Al)
Silver (Ag)
Gold (Au)
Chrome (Cr)
Copper (Cu)
Indium (In)
Titanium (Ti)
Iron (Fe)
Nickel (Ni)
Lead (Pb)
Palladium (Pd)
Platine (Pt)
Antimon (Sb)
Tungsten (W)
Semiconductors
C [100]
GaAs
Ge
InSb
Si(c)
Poly-Si
Silicon (single-crystal)
Insulators
Al_2O_3
SiC (6H)
Si_3N_4
SiO_2
ZnO
Borosilicate
Polymers
Nylon
PDMA
PMMA
Polymide
Polyethylene
PTFE
PVC

Piezoelectric Materials Database

The Piezoelectric materials database included with this module contains the following materials:

MATERIAL
Barium Sodium Niobate
Barium Titanate
Barium Titanate (poled)
Lithium Niobate
Lithium Tantalate
Lead Zirconate Titanate (PZT-2)
Lead Zirconate Titanate (PZT-4)
Lead Zirconate Titanate (PZT-4D)
Lead Zirconate Titanate (PZT-5A)
Lead Zirconate Titanate (PZT-5H)
Lead Zirconate Titanate (PZT-5J)
Lead Zirconate Titanate (PZT-7A)
Lead Zirconate Titanate (PZT-8)
Quartz
Rochelle Salt
Bismuth Germanate
Cadmium Sulfide
Gallium Arsenide
Tellurium Dioxide
Zinc Oxide
Zinc Sulfide
Ammonium Dihydrogen Phosphate
Aluminum Nitride

All materials define the following material properties needed for piezoelectric modeling:

MATERIAL PROPERTY	DESCRIPTION
c_E	Elasticity matrix
e	Coupling matrix, stress-charge
$\varepsilon_{\mathrm{r}S}$	Relative permittivity, stress-charge
s_E	Compliance matrix
d	Coupling matrix, strain-charge
$\varepsilon_{\mathrm{r}T}$	Relative permittivity, strain-charge
ρ	Density

Piezoresistivity Materials Database

The Piezoresistivity materials database is included with the MEMS Module and contains the following materials:

MATERIAL
p-Silicon (single-crystal, lightly doped)
n-Silicon (single-crystal, lightly doped)
p-Silicon (polycrystalline, lightly doped)
n-Silicon (polycrystalline, lightly doped)

All materials define the following material properties needed for modeling the piezoresistance effect:

MATERIAL PROPERTY	DESCRIPTION
D	Elasticity matrix
DVo	Elasticity matrix, Voigt notation
ρ	Density
ε_r	Relative permittivity
σ	Electrical conductivity
Π	Piezoresistive coupling matrix
m_l	Elastoresistive coupling matrix

Both the electrical conductivity and the piezoresistive or elastoresistive coupling matrix are strong functions of the material dopant density. The material models include appropriate functions, although the piezoresistive and elastoresistive matrices scale only with the conductivity, which is appropriate only at lower dopant densities (below approximately 10^{16} cm^{-3}). The low doping level piezoresistance and elastoresistance values are based on those given in Ref. 1. The conductivity is computed from an empirical functional fit to experimental data given in equation 8 of Ref. 2. Data on the piezoresistance properties of Silicon at higher doping levels is available in Ref. 3 and Ref. 4. Because this data does not include all components of the coupling matrix, it is not included in the material models.

The dopant density must be entered for the material as a model input in the piezoresistive or conductive material node. It can be entered as a constant value or as an expression (for example, a spatially varying function could be used).

Batteries and Fuel Cells Materials Database

The Batteries and Fuel Cells Materials database is included with the Batteries & Fuel Cells Module and contains the materials listed in Table 9-16. The material property groups (including all associated properties) are listed in Table 9-4.

TABLE 9-16: BATTERIES & FUEL CELLS MODULE MATERIALS DATABASE

MATERIAL	INTENDED USE
Sulfuric Acid (Lead-Acid Battery)	Electrolyte
Pb Electrode (Negative, Lead-Acid Battery)	Equilibrium potential
PbO2 Electrode (Positive, Lead-Acid Battery)	Equilibrium potential
LixC6 Electrode (Negative, Li-ion Battery)	Equilibrium potential
LixMn2O4 Electrode (Positive, Li-ion Battery)	Equilibrium potential
LixCoO2 Electrode (Positive, Li-ion Battery)	Equilibrium potential
HxLiN5 Electrode (Negative, NiMH Battery)	Equilibrium potential
NiOHO-Hx Electrode (Positive discharge, NiMH Battery)	Equilibrium potential

TABLE 9-16: BATTERIES & FUEL CELLS MODULE MATERIALS DATABASE

MATERIAL	INTENDED USE
NiOHO-Hx Electrode (Positive charge, NiMH Battery)	Equilibrium potential
1:2 EC:DMC/LiPF6 (Li-ion Battery)	Electrolyte
2:1 EC:DMC/LiPF6 (Li-ion Battery)	Electrolyte
1:1 EC:DEC/LiPF6 (Li-ion Battery)	Electrolyte

Semiconductor Materials Database

The Semiconductor Materials database is included with the Semiconductor Module and contains the materials listed in Table 9-17. The material property groups (including all associated properties) are listed in Table 9-13.

The Silicon material contains parameters for all the material property groups, while the other materials contain only the material parameters in the Semiconductor and Basic material groups.

Table 9-13 also gives the references used for the silicon material properties. The material properties for materials other than Silicon are obtained from Ref. 9 and Ref. 10.

TABLE 9-17: SEMICONDUCTOR MODULE MATERIALS DATABASE

MATERIAL
Si
Ge
GaAs
Al(x)Ga(1-x)As
GaP
GaSb
InAs
InP
InSb

References for the Material Databases

REFERENCES FOR THE LIQUIDS AND GASES MATERIAL DATABASE

1. *ASHRAE Handbook of Fundamentals*, American Society of Heating, Refrigerating and Air Conditioning Engineers, 1993.

2. E.R.G. Eckert and M. Drake, Jr., *Analysis of Heat and Mass Transfer*, Hemisphere Publishing, 1987.

3. H. Kashiwagi, T. Hashimoto, Y. Tanaka, H. Kubota, and T. Makita, “Thermal Conductivity and Density of Toluene in the Temperature Range 273-373K at Pressures up to 250 MPa,” *Int. J. Thermophys.*, vol. 3, no. 3, pp. 201–215, 1982.

4. C.A. Nieto de Castro, S.F.Y. Li, A. Nagashima, R.D. Trengove, and W.A. Wakeham, “Standard Reference Data for the Thermal Conductivity of Liquids,” *J. Phys. Chem. Ref. Data*, vol. 15, no. 3, pp. 1073–1086, 1986.

5. B.E. Poling, J.M. Prausnitz, and J.P. O’Connell, *The Properties of Gases and Liquids*, 5th ed., McGraw-Hill, 2001.

6. C.F. Spencer and B.A. Adler, “A Critical Review of Equations for Predicting Saturated Liquid Density,” *J. Chem. Eng. Data*, vol. 23, no. 1, pp. 82–88, 1978.

7. N.B.Vargnaftik, *Tables of Thermophysical Properties of Liquids and Gases*, 2nd ed., Hemisphere Publishing, 1975.

8. R.C.Weast (ed.), *CRC Handbook of Chemistry and Physics*, 69th ed., CRC Press, 1988.

9. M. Zabransky and V. Ruzicka, Jr., “Heat Capacity of Liquid n-Heptane Converted to the International Temperature Scale of 1990,” *Phys. Chem. Ref. Data*, vol. 23, no. 1, pp. 55–61, 1994.

10. M. Zabransky, V. Ruzicka, Jr., and E.S. Domalski, “Heat Capacity of Liquids: Critical Review and Recommended Values. Supplement I,” *J. Phys. Chem. Ref. Data*, vol. 30, no. 5, pp. 1199–1397, 2002.

REFERENCES FOR THE MEMS MATERIALS DATABASE

1. J.W. Gardner, V.K. Varadan, and O.O. Awadelkarim, *Microsensors, MEMS, and Smart Devices*, John Wiley & Sons, 2001.

2. D.R. Lide (ed.), *CRC Handbook of Chemistry and Physics*, 84th edition, CRC Press, 2003.

3. A.M. James and M.P. Lord, *MacMillan’s Chemical and Physical Data*, MacMillan’s Press, 1992.

4. M. Gad-el-Hak (ed.), *The MEMS Handbook*, CRC Press, 2002.

5. *New Semiconductor Materials. Characteristics and Properties*, www.ioffe.ru/SVA/NSM, 2003.

6. *Ceramics WebBook*, www.ceramics.nist.gov/srd/scd/scdquery.htm, 2003.

7. J.E. Mark, *The Polymer Data Handbook*, 2nd edition, Oxford University Press, 2009.

8. Lotters et. al, “The mechanical properties of the rubber elastic polymer polydimethylsiloxane for sensor applications,” *J. Micromech Microeng*, vol. 7 145–147, 1997.

REFERENCES FOR THE PIEZORESISTIVITY MATERIALS DATABASE

1. C.S. Smith, “Piezoresistance Effect in Germanium and Silicon,” *Physical Review*, vol. 94, no. 1, pp. 42–49, 1957.

2. C. Jacoboni, C. Canali, G. Ottaviani, and A. Alberigi Quaranta, “A Review of Some Charge Transport Properties of Silicon,” *Solid-State Electronics*, vol. 20, pp. 77–89, 1977.

3. O.N. Tufte and E.L. Stelzer, “Piezoresistance Properties of Heavily Doped n-Type Silicon,” *Physical Review*, vol. 133, no. 6A, pp. A1705–A1716, 1964.

4. O.N. Tufte and E.L Stelzer, “Piezoresistive Properties of Silicon Diffused Layers,” *J. Applied Physics*, vol. 34, no. 2, pp. 313–318, 1963.

REFERENCES FOR THE SEMICONDUCTOR MATERIALS DATABASE

1. M. Shur, *Physics of Semiconductor Devices*, Prentice Hall, 1990.

2. S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer Verlag, 1984.

3. Y. Okuto and C. R. Crowell, “Threshold energy effect on avalanche breakdown voltage in semiconductor junctions”, *Solid State Electronics*, vol. 18, pp. 161-168, 1975.

4. Approximate values are given, based loosely on Ref. 2. In practice these values should depend on the local doping, as discussed in Ref. 2. Use the user-defined option to implement this functionality.

5. N.D. Arora, J.R. Hauser, and D.J. Roulston, “Electron and Hole Mobilities in Silicon as a Function of Concentration and Temperature,” *IEEE Transactions on Electron Devices*, vol. 29, no. 2, pp. 292–295, 1982.

6. C. Canali, G. Majni, R. Minder and G. Ottaviani “Electron and Hole Drift Velocity Measurements in Silicon and Their Empirical Relation to Electric Field and Temperature”, *IEEE Transactions on Electron Devices*, vol. 22, no. 11, pp. 1045– 1047, 1975. Note the correction in: G. Ottaviani, “Correction to ‘Electron and hole drift velocity measurements in silicon and their empirical relation to electric field and temperatures’”, *IEEE Transactions on Electron Devices*, vol. 23, no. 9, pp. 1113, 1976.

7. J.M. Dorkel and Ph. Leturcq, “Carrier Mobilities in Silicon Semi-empirically Related to Temperature, Doping and Injection Level,” *Solid-State Electronics*, vol. 24, no. 9, pp. 821–825, 1981.

8. C. Lombardi, S. Manzini, A. Saporito and M. Vanzi, “A Physically Based Mobility Model for Numerical Simulation of Nonplanar Devices”, IEEE Transactions on Computer-Aided Design, vol. 7, no. 11, 1988.

9. M. Levinshtein, S. Rumyantsev and M. Shur, *Handbook Series on Semiconductor Parameters, Volume 1: Si, Ge, C (Diamond), GaAs, GaP, InAs, InP, InSb*, World Scientific, 2000.

10. M. Levinshtein, S. Rumyantsev and M. Shur, *Handbook Series on Semiconductor Parameters, Volume II: Ternary and Quaternary III-V compounds*, World Scientific, 2000.

REFERENCES FOR THE BIOHEAT MATERIALS DATABASE

1. S. Jacques, S. Rastegar, S. Thomsen, and M. Motamedi, “Nonlinear Finite-element Analysis The Role of Dynamic Changes in Blood Perfusion and Optical Properties in Laser Coagulation of Tissue,” *IEEE J. Selected Topics in Quantum Electronics*, vol. 2, no. 4, pp. 922–933, 1996.

2. P. Prakash and C.J. Diederich, “Considerations for theoretical modeling of thermal ablation with catheter-based ultrasonic sources: implications for treatment planning, monitoring and control,” *Int. J. Hyperthermia*, vol. 28, no. 1, pp. 69–86, 2012 (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3366914/).

3. F. Xu, K.A. Seffen and T.J. Lu, “Temperature-Dependent Mechanical Behaviors of Skin Tissue,” *IAENG Int. J. Computer Science*, vol. 35, no 1, 2008 (.http://www.iaeng.org/IJCS/issues_v35/issue_1/IJCS_35_1_13.pdf).

4. http://www.ncbi.nlm.nih.gov/pubmed/12953912/

5. http://www.itis.ethz.ch/itis-for-health/tissue-properties/database/database-summary/

10

The AC/DC Interfaces

This chapter explains the interfaces for modeling electromagnetics, which you find under the **AC/DC** branch (⚡) when adding a physics interface. It also contains sections about general fundamentals and theory for electric fields.

The Electromagnetics Interfaces

For simulating electromagnetic fields, COMSOL Multiphysics has three physics interfaces.

With the first two, you can perform static simulations to solve for electric properties:

- Electrostatics
- Electric Currents
- Magnetic Fields

These interfaces are available in all space dimensions.

For 2D axisymmetric and 2D geometries, you can use the Magnetic Fields interface to perform magnetostatic simulations.

This section begins with a brief introduction to electromagnetics and a definition of the electromagnetic quantities. Then it describes each of the physics interfaces in detail.

The optional AC/DC Module contains specialized and extended interfaces for electromagnetic simulations, for example, for computations of inductors and capacitors. The optional RF Module includes interfaces for simulating electromagnetic wave propagation that are especially useful in microwave engineering and photonics.

Fundamentals of Electromagnetics

Maxwell's Equations

The problem of electromagnetic analysis on a macroscopic level is that of solving *Maxwell's equations* subject to certain boundary conditions. Maxwell's equations are a set of equations, written in differential or integral form, stating the relationships between the fundamental electromagnetic quantities. These quantities are:

- Electric field intensity $\mathbf{E}$
- Electric displacement or electric flux density $\mathbf{D}$
- Magnetic field intensity $\mathbf{H}$
- Magnetic flux density $\mathbf{B}$
- Current density $\mathbf{J}$
- Electric charge density ρ

The equations can be formulated in differential form or integral form. The differential form is presented here because it leads to differential equations that the finite element method can handle. For general time-varying fields, Maxwell's equations can be written as:

$$\begin{aligned}\nabla\times\mathbf{H} &= \mathbf{J}+\frac{\partial\mathbf{D}}{\partial t}\\ \nabla\times\mathbf{E} &= -\frac{\partial\mathbf{B}}{\partial t}\\ \nabla\cdot\mathbf{D} &= \rho\\ \nabla\cdot\mathbf{B} &= 0\end{aligned}$$

The first two equations are also referred to as *Maxwell-Ampère's law* and *Faraday's law*, respectively. Equation three and four are two forms of *Gauss' law*: the electric and magnetic form, respectively.

Another fundamental equation is the *equation of continuity*

$$\nabla\cdot\mathbf{J} = -\frac{\partial\rho}{\partial t}$$

Out of the five equations mentioned, only three are independent. The first two combined with either the electric form of Gauss' law or the equation of continuity form such an independent system.

Constitutive Relations

To obtain a closed system, the equations include *constitutive relations* that describe the macroscopic properties of the medium. They are given as

$$\begin{aligned}\mathbf{D} &= \varepsilon_0\mathbf{E}+\mathbf{P}\\ \mathbf{B} &= \mu_0(\mathbf{H}+\mathbf{M})\\ \mathbf{J} &= \sigma\mathbf{E}\end{aligned} \qquad (10\text{-}1)$$

where ε_0 is the permittivity of vacuum, μ_0 is the permeability of vacuum, and σ the electrical conductivity. In the SI system, the permeability of vacuum is chosen to be $4\pi\cdot10^{-7}$ H/m. The velocity of an electromagnetic wave in a vacuum is given as c_0 and the permittivity of a vacuum is derived from the relation:

$$\varepsilon_0 = \frac{1}{c_0^2\mu_0} = 8.854 \cdot 10^{-12}\ \mathrm{F/m} \approx \frac{1}{36\pi} \cdot 10^{-9}\ \mathrm{F/m}$$

The electromagnetic constants ε_0, μ_0, and c_0 are available in COMSOL Multiphysics as predefined physical constants.

The *electric polarization vector* $\mathbf{P}$ describes how the material is polarized when an electric field $\mathbf{E}$ is present. It can be interpreted as the volume density of *electric dipole* moments. $\mathbf{P}$ is generally a function of $\mathbf{E}$. Some materials can have a nonzero $\mathbf{P}$ also when there is no electric field present.

The *magnetization vector* $\mathbf{M}$ similarly describes how the material is magnetized when a magnetic field $\mathbf{H}$ is present. It can be interpreted as the volume density of *magnetic dipole* moments. $\mathbf{M}$ is generally a function of $\mathbf{H}$. Permanent magnets, for instance, have a nonzero $\mathbf{M}$ also when there is no magnetic field present.

For linear materials, the polarization is directly proportional to the electric field, $\mathbf{P} = \varepsilon_0\chi_\mathrm{e}\mathbf{E}$, where χ_e is the electric susceptibility. Similarly in linear materials, the magnetization is directly proportional to the magnetic field, $\mathbf{M} = \chi_\mathrm{m}\mathbf{H}$, where χ_m is the magnetic susceptibility. For such materials, the constitutive relations are:

$$\mathbf{D} = \varepsilon_0(1+\chi_\mathrm{e})\mathbf{E} = \varepsilon_0\varepsilon_\mathrm{r}\mathbf{E} = \varepsilon\mathbf{E}$$
$$\mathbf{B} = \mu_0(1+\chi_\mathrm{m})\mathbf{H} = \mu_0\mu_\mathrm{r}\mathbf{H} = \mu\mathbf{H}$$

The parameter ε_r is the relative permittivity and μ_r is the relative permeability of the material. Usually these are scalar properties but can, in the general case, be 3-by-3 tensors when the material is anisotropic. The properties ε and μ (without subscripts) are the permittivity and permeability of the material, respectively.

GENERALIZED CONSTITUTIVE RELATIONS

> The Charge Conservation node describes the macroscopic properties of the medium (relating the electric displacement $\mathbf{D}$ with the electric field $\mathbf{E}$) and the applicable material properties.

For nonlinear materials, a generalized form of the constitutive relationships is useful. The relationship used for electric fields is $\mathbf{D} = \varepsilon_0\varepsilon_\mathrm{r}\mathbf{E} + \mathbf{D}_\mathrm{r}$ where $\mathbf{D}_\mathrm{r}$ is the *remanent displacement*, which is the displacement when no electric field is present.

Similarly, a generalized form of the constitutive relation for the magnetic field is

$$\mathbf{B} = \mu_0\mu_\mathrm{r}\mathbf{H} + \mathbf{B}_\mathrm{r}$$

where $\mathbf{B}_\mathrm{r}$ is the *remanent magnetic flux density*, which is the magnetic flux density when no magnetic field is present.

For some materials, there is a nonlinear relationship between $\mathbf{B}$ and $\mathbf{H}$ such that

$$\mathbf{B} = f(|\mathbf{H}|)$$

The relation defining the current density is generalized by introducing an externally generated current $\mathbf{J}_\mathrm{e}$. The resulting constitutive relation is $\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_\mathrm{e.}$

Potentials

Under certain circumstances it can be helpful to formulate the problems in terms of the electric scalar potential V and the magnetic vector potential $\mathbf{A}$. They are given by the equalities:

$$\mathbf{B} = \nabla \times \mathbf{A}$$

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t}$$

The defining equation for the magnetic vector potential is a direct consequence of the magnetic Gauss' law. The electric potential results from Faraday's law.

Material Properties

Until now, there has only been a formal introduction of the constitutive relations. These seemingly simple relations can be quite complicated at times. There are four main groups of materials where they require some consideration. A given material can belong to one or more of these groups. The groups are:

- Inhomogeneous Materials
- Anisotropic Materials
- Nonlinear Materials
- Dispersive Materials

INHOMOGENEOUS MATERIALS

Inhomogeneous materials are the least complicated. An inhomogeneous medium is one in which the constitutive parameters vary with the space coordinates so that different field properties prevail at different parts of the material structure.

ANISOTROPIC MATERIALS

For anisotropic materials the field relationships at any point differ for different directions of propagation. This means that a 3-by-3 tensor is necessary to properly define the constitutive relationships. If this tensor is symmetric, the material is often referred to as *reciprocal*. In such cases, rotate the coordinate system such that a diagonal matrix results. If two of the diagonal entries are equal, the material is *uniaxially anisotropic*. If none of the elements has the same value, the material is *biaxially anisotropic* (Ref. 2). Anisotropic parameters are needed, for example, to examine permittivity in crystals (Ref. 2) and when working with conductivity in solenoids.

NONLINEAR MATERIALS

Nonlinearity is the effect of variations in permittivity or permeability with the intensity of the electromagnetic field. Nonlinearity also includes hysteresis effects, where not only the current field intensities influence the physical properties of the material, but also the history of the field distribution.

DISPERSIVE MATERIALS

Dispersion describes changes in a wave's velocity with wavelength. In the frequency domain, dispersion is expressed with a frequency dependence of the constitutive laws.

About the Boundary and Physics Interface Conditions

To get a full description of an electromagnetics problem, boundary conditions must be specified at material interfaces and physical boundaries. At interfaces between two media, the boundary conditions can be expressed mathematically as

$$\mathbf{n}_2 \times (\mathbf{E}_1 - \mathbf{E}_2) = \mathbf{0}$$

$$\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

$$\mathbf{n}_2 \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s$$

$$\mathbf{n}_2 \cdot (\mathbf{B}_1 - \mathbf{B}_2) = 0$$

where ρ_s and $\mathbf{J}_s$ denote surface charge density and surface current density, respectively, and $\mathbf{n}_2$ is the outward normal from medium two. Of these four conditions, only two are independent. This is an overdetermined system of equations, so it needs to be reduced. First select either equation one or equation four. Then select either equation two or equation three. Together these selections form a set of two independent conditions.

From these relationships, the interface condition is derived for the current density,

$$\mathbf{n}_2 \cdot (\mathbf{J}_1 - \mathbf{J}_2) = -\frac{\partial \rho_s}{\partial t}$$

INTERFACE BETWEEN A DIELECTRIC AND A PERFECT CONDUCTOR

A perfect conductor has infinite electrical conductivity and thus no internal electric field. Otherwise, it would produce an infinite current density according to the third fundamental constitutive relation. At an interface between a dielectric and a perfect conductor, the boundary conditions for the **E** and **D** fields are simplified. Assume that subscript 1 corresponds to a perfect conductor; then $\mathbf{D}_1 = \mathbf{0}$ and $\mathbf{E}_1 = \mathbf{0}$ in the relationships just given. If it is a time-varying case, then $\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{H}_1 = \mathbf{0}$, as well, as a consequence of Maxwell's equations. The result is the following set of boundary conditions for the fields in the dielectric medium for the time-varying case:

$$\begin{aligned} -\mathbf{n}_2 \times \mathbf{E}_2 &= 0 \\ -\mathbf{n}_2 \times \mathbf{H}_2 &= \mathbf{J}_s \\ -\mathbf{n}_2 \cdot \mathbf{D}_2 &= \rho_s \\ -\mathbf{n}_2 \cdot \mathbf{B}_2 &= 0 \end{aligned}$$

Electromagnetic Forces

The Magnetic Field interface contains a predefined domain-level variable for calculating the Lorentz force, which gives the force distribution exerted on a current-carrying conductor placed in magnetic flux density **B**. The Lorentz force is defined as $\mathbf{F} = \mathbf{J} \times \mathbf{B}$.

The Lorentz force gives very good accuracy for electromagnetic force calculations in conducting domains. For nonconducting domains, use a more general method— integrating the Maxwell stress tensor variables over the boundaries of the object for which to calculate the total force. The Maxwell surface stress tensor is available as a boundary variable.

References for Electromagnetic Theory

1. D.K. Cheng, *Field and Wave Electromagnetics*, Addison-Wesley, Reading, Massachusetts, 1989.

2. J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 1993.

3. B.D. Popovic, *Introductory Engineering Electromagnetics*, Addison-Wesley, Reading, Massachusetts, 1971.

Theory of Electric Fields

COMSOL Multiphysics includes physics interfaces for the modeling of static electric fields and currents. Deciding what specific physics interface and study type to select for a particular modeling situation requires a basic understanding of the charge dynamics in conductors. This section is a brief introduction to Charge Relaxation Theory.

Physics interfaces for the modeling of dynamic, quasi-static (that is, without including wave propagation effects) electric fields and currents are available with the AC/DC Module and MEMS Module.

Charge Relaxation Theory

The different physics interfaces involving only the scalar electric potential can be interpreted in terms of the charge relaxation process. The fundamental equations involved are *Ohm's law*

$$\mathbf{J} = \sigma\mathbf{E}$$

the *equation of continuity*

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\mathbf{J} = 0$$

and *Gauss' law*

$$\nabla\cdot(\varepsilon\mathbf{E}) = \rho$$

By combining these, one can deduce the following differential equation for the space charge density in a homogeneous medium

$$\frac{\partial\rho}{\partial t} + \frac{\sigma}{\varepsilon}\rho = 0$$

This equation has the solution

$$\rho(t) = \rho_0 e^{-t/\tau}$$

where

$$\tau = \frac{\varepsilon}{\sigma}$$

is called the charge relaxation time. For a good conductor like copper, τ is of the order of 10^{-19} s whereas for a good insulator like silica glass, it is of the order of 10^3 s. For a pure insulator, it becomes infinite.

When modeling real-world devices, there is not only the *intrinsic time scale* of the charge relaxation time but also an *external time scale* t at which a device is energized or the observation time. It is the relation between the external time scale and the charge relaxation time that determines what physics interface and study type to use. The results are summarized in Table 10-1 below,

TABLE 10-1: SUITABLE INTERFACE AND STUDY TYPE FOR DIFFERENT TIME-SCALE REGIMES.

CASE	INTERFACE	STUDY TYPE
$\tau >> t$	Electrostatics	Stationary

TABLE 10-1: SUITABLE INTERFACE AND STUDY TYPE FOR DIFFERENT TIME-SCALE REGIMES.

CASE	INTERFACE	STUDY TYPE
τ<<t	Electric Currents	Stationary
τ~t	Electric Currents	Time Dependent or, with the AC/DC Module, MEMS Module, or Semiconductor Module, Frequency Domain

FIRST CASE: τ >> T

If the external time scale is short compared to the charge relaxation time, the charges do not have time to redistribute to any significant degree.Thus the charge distribution can be considered as a given model input, and the best approach is to solve the Electrostatics formulation using the electric potential V.

By combining the definition of the potential with Gauss' law, you can derive the classical Poisson's equation. Under static conditions, the electric potential V is defined by the equivalence $\mathbf{E} = -\nabla V$. Using this together with the constitutive relation $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$ between $\mathbf{D}$ and $\mathbf{E}$, you can rewrite Gauss' law as a variant of Poisson's equation

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

This equation is used in the Electrostatics interface. It is worth noting that Gauss' law does not require the charge distribution to be static. Thus, provided dynamics are slow enough that induced electric fields can be neglected and hence a scalar electric potential is justified, the formulation can be used also in the Time Dependent study type. That typically involves either prescribing the charge dynamics or coupling a separate formulation for this.

Such separate charge transport formulations can be found in the Plasma Module and the Chemical Reaction Engineering Module.

SECOND CASE: τ <<T

If the external time scale is long compared to the charge relaxation time, the stationary solution to the equation of continuity has been reached. In a stationary coordinate system, a slightly more general form than above of Ohm's law states that

$$\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}^{\mathrm{e}}$$

where $\mathbf{J}^{\mathrm{e}}$ is an externally generated current density. The static form of the equation of continuity then reads

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma \nabla V - \mathbf{J}^{\mathrm{e}}) = 0$$

To handle current sources the equation can be generalized to

$$-\nabla \cdot (\sigma \nabla V - \mathbf{J}^{\mathrm{e}}) = Q_j$$

This equation is used in the static study type for the Electric Currents interface.

Theory of Electrostatics

The Electrostatics Interface is available for 3D, 2D in-plane, and 2D axisymmetric models. Applications with electrostatic equations include high-voltage apparatus, electronic devices, and capacitors. The term “statics” is not to be interpreted literally—it is the observation time or time scale at which the applied excitation changes is short compared to the charge relaxation time and that the electromagnetic wavelength and skin depth are very large compared to the size of the domain of interest.

If you do not know whether to use the Electric Currents or the Electrostatics interface, which both solve for the scalar electric potential V, consider using an explicit charge transport model. See Charge Relaxation Theory.

Electrostatics Equations

Under static conditions the electric potential, V, is defined by the relationship:

$$\mathbf{E} = -\nabla V$$

Combining this equation with the constitutive relationship $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$ between the electric displacement $\mathbf{D}$ and the electric field $\mathbf{E}$, it is possible to represent Gauss’ law as the following equation:

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

In this equation, the physical constant, ε_0 (SI unit: F/m) is the permittivity of vacuum, $\mathbf{P}$ (SI unit: $\mathrm{C/m^2}$) is the electric polarization vector, and ρ (SI unit: $\mathrm{C/m^3}$) is a space charge density. This equation describes the electrostatic field in dielectric materials.

For in-plane 2D modeling, the Electrostatics interface assumes a symmetry where the electric potential varies only in the x and y directions and is constant in the z direction. This implies that the electric field, $\mathbf{E}$, is tangential to the xy-plane. With this symmetry, the same equation is solved as in the 3D case. The interface solves the following equation where d is the thickness in the z direction:

$$-\nabla \cdot d(\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

The axisymmetric version of the physics interface considers the situation where the fields and geometry are axially symmetric. In this case the electric potential is constant in the φ direction, which implies that the electric field is tangential to the rz-plane.

The Electrostatics Interface in Time Dependent or Frequency Domain Studies

The Electrostatics Interface can be solved also in a dynamic study (Time Dependent or Frequency Domain). The equation system solved, however, is always the one presented in the previous section for the stationary case, in which no transient electromagnetic effects are taken into account. The difference is that the sources of the problem (charge densities, electric potential) are assumed to be time-varying (in a Time Dependent study) or time-harmonic (in a Frequency Domain study). The support for dynamic studies simplifies the coupling of the Electrostatics interface with other physics. Using the interface in a dynamic study is a valid approximation only if the time-scale (or the frequency) of the study is so slow that transient electromagnetic effects can be neglected; for example, in acoustic or structural problems.

The Electrostatics interface also supports the small-signal analysis study sequence, that can be used when on a static bias charge or voltage is superposed a time-harmonic perturbation.

Theory of Electric Currents

The Electric Currents Interface solves a current conservation problem for the scalar electric potential V and is available for 3D, 2D in-plane, and 2D axisymmetric models. Electrolysis and the computation of resistances of grounding plates are examples that involve conductive media with electrical conductivities and electric currents. If you are uncertain whether to use the Electric Currents or the Electrostatics interface, which both solve for the scalar electric potential V, refer to the section on Charge Relaxation Theory.

Electric Currents Equations in Steady State

When handling stationary electric currents in conductive media you must consider the stationary equation of continuity. In a stationary coordinate system, the point form of *Ohm's law* states that:

$$\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_\mathrm{e}$$

where σ is the electrical conductivity (SI unit: S/m), and $\mathbf{J}_\mathrm{e}$ is an externally generated current density (SI unit: A/m^2). The static form of the equation of continuity then states:

$$\nabla\cdot\mathbf{J} = -\nabla\cdot(\sigma\nabla V - \mathbf{J}_\mathrm{e}) = 0$$

To handle current sources, you can generalize the equation to:

$$-\nabla\cdot(\sigma\nabla V - \mathbf{J}_\mathrm{e}) = Q_j$$

In planar 2D the Electric Currents interface assumes that the model has a symmetry where the electric potential varies only in the x and y directions and is constant in the z direction. This implies that the electric field, $\mathbf{E}$, is tangential to the xy-plane. The Electric Currents interface then solves the following equation where d is the thickness in the z direction:

$$-\nabla\cdot d(\sigma\nabla V - \mathbf{J}_\mathrm{e}) = dQ_j \qquad (10\text{-}2)$$

In 2D axisymmetry, the Electric Currents interface considers the situation where the fields and geometry are axially symmetric. In this case the electric potential is constant in the φ direction, which implies that the electric field is tangential to the rz-plane.

Theory of Magnetic and Electric Fields

Quasi-static analysis of magnetic and electric fields is valid under the assumption that $\partial\mathbf{D}/\partial t = 0$.

Maxwell's Equations

This implies that it is possible to rewrite Maxwell's equations in the following manner:

$$\nabla\times\mathbf{H} = \mathbf{J} = \sigma(\mathbf{E}+\mathbf{v}\times\mathbf{B})+\mathbf{J}_\mathrm{e}$$

$$\nabla\times\mathbf{E} = -\frac{\partial\mathbf{B}}{\partial t}$$

$$\nabla\cdot\mathbf{B} = 0$$

$$\nabla\cdot\mathbf{D} = \rho$$

$$\nabla\cdot\mathbf{J} = 0$$

Here $\mathbf{J}_\mathrm{e}$ is an externally generated current density and $\mathbf{v}$ is the velocity of the conductor. The crucial criterion for the quasi-static approximation to be valid is that the currents and the electromagnetic fields vary slowly. This means that the dimensions of the structure in the problem need to be small compared to the wavelength.

Magnetic and Electric Potentials

Using the definitions of the potentials,

$$\mathbf{B} = \nabla\times\mathbf{A}$$

$$\mathbf{E} = -\nabla V-\frac{\partial\mathbf{A}}{\partial t}$$

and the constitutive relation $\mathbf{B}=\mu_0(\mathbf{H}+\mathbf{M})$, Ampère's law can be rewritten as

$$\sigma\frac{\partial\mathbf{A}}{\partial t}+\nabla\times(\mu_0^{-1}\nabla\times\mathbf{A}-\mathbf{M})-\sigma\mathbf{v}\times(\nabla\times\mathbf{A})+\sigma\nabla V = \mathbf{J}_\mathrm{e} \qquad (10\text{-}3)$$

The equation of continuity, which is obtained by taking the divergence of the above equation, adds the following equation:

$$\nabla\cdot\left(-\sigma\frac{\partial\mathbf{A}}{\partial t}+\sigma\mathbf{v}\times(\nabla\times\mathbf{A})-\sigma\nabla V+\mathbf{J}_\mathrm{e}\right) = 0 \qquad (10\text{-}4)$$

Equation 10-3 and Equation 10-4 form a system of equations for the two potentials $\mathbf{A}$ and V.

Gauge Transformations

The electric and magnetic potentials are not uniquely defined from the electric and magnetic fields through

$$\mathbf{E} = -\frac{\partial\mathbf{A}}{\partial t}-\nabla V$$

$$\mathbf{B} = \nabla\times\mathbf{A}$$

Introducing two new potentials

$$\tilde{\mathbf{A}} = \mathbf{A} + \nabla\Psi$$
$$\tilde{V} = V - \frac{\partial\Psi}{\partial t}$$

gives the same electric and magnetic fields:

$$\mathbf{E} = -\frac{\partial\mathbf{A}}{\partial t} - \nabla V = -\frac{\partial(\tilde{\mathbf{A}} - \nabla\Psi)}{\partial t} - \nabla\left(\tilde{V} + \frac{\partial\Psi}{\partial t}\right) = -\frac{\partial\tilde{\mathbf{A}}}{\partial t} - \nabla\tilde{V}$$
$$\mathbf{B} = \nabla\times\mathbf{A} = \nabla\times(\tilde{\mathbf{A}} - \nabla\Psi) = \nabla\times\tilde{\mathbf{A}}$$

The variable transformation of the potentials is called a *gauge transformation*. To obtain a unique solution, choose the gauge, that is, put constraints on Ψ that make the solution unique. Another way of expressing this additional condition is to put a constraint on $\nabla\cdot\mathbf{A}$. A vector field is uniquely defined up to a constant if both $\nabla\cdot\mathbf{A}$ and $\nabla\times\mathbf{A}$ are given. This is called *Helmholtz's theorem*.

One particular gauge is the *Coulomb gauge* given by the constraint: $\nabla\cdot\mathbf{A} = 0$.

Selecting a Particular Gauge

Important observations are that in the dynamic case **A** and *V* are coupled via the selected gauge. For a dynamic formulation, it is also possible to select a Ψ such that the scalar electric potential vanishes and only the magnetic vector potential has to be considered. The dynamic formulations (Frequency Domain and Time Dependent study types) of the Magnetic Fields interface are operated in this gauge as it involves only **A**. In the static limit, **A** and *V* are not coupled via the gauge selection and thus any gauge can be chosen for **A** when performing magnetostatic modeling.

The Gauge and the Equation of Continuity for Dynamic Fields

After eliminating the electric potential by choosing the appropriate gauge and disregarding the velocity term. The equation of continuity obtained by taking the divergence of Ampère's law reads:

$$\nabla\cdot\left(-\sigma\frac{\partial\mathbf{A}}{\partial t} + \mathbf{J}_{\mathrm{e}}\right) = 0$$

It is clear that unless the electrical conductivity is uniform, the particular gauge used to eliminate *V* cannot be the Coulomb gauge as that would violate the equation of continuity and would thereby also violate Ampère's law.

Time-Harmonic Magnetic Fields

In the time-harmonic case, there is no computational cost for including the displacement current in Ampère's law (then called Maxwell-Ampère's law):

$$\nabla\times\mathbf{H} = \mathbf{J} = \sigma(\mathbf{E} + \mathbf{v}\times\mathbf{B}) + j\omega\mathbf{D} + \mathbf{J}^{\mathrm{e}}$$

In the transient case the inclusion of this term leads to a second-order equation in time, but in the harmonic case there are no such complications. Using the definition of the electric and magnetic potentials, the system of equations becomes:

$$-\nabla\cdot((j\omega\sigma - \omega^2\varepsilon_0)\mathbf{A} - \sigma\mathbf{v}\times(\nabla\times\mathbf{A}) + (\sigma + j\omega\varepsilon_0)\nabla V - (\mathbf{J}^{\mathrm{e}} + j\omega\mathbf{P})) = 0$$
$$(j\omega\sigma - \omega^2\varepsilon_0)\mathbf{A} + \nabla\times(\mu_0^{-1}\nabla\times\mathbf{A} - \mathbf{M}) - \sigma\mathbf{v}\times(\nabla\times\mathbf{A}) + (\sigma + j\omega\varepsilon_0)\nabla V = \mathbf{J}^{\mathrm{e}} + j\omega\mathbf{P}$$

The constitutive relation $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$ has been used for the electric field.

To obtain a particular gauge that reduces the system of equation, choose $\Psi = -jV/\omega$ in the gauge transformation. This gives:

$$\tilde{\mathbf{A}} = \mathbf{A} - \frac{j}{\omega}\nabla V \qquad\qquad \tilde{V} = 0$$

When $\tilde{V}$ vanishes from the equations, only the second one is needed,

$$(j\omega\sigma - \omega^2\varepsilon_0)\tilde{\mathbf{A}} + \nabla\times(\mu_0^{-1}\nabla\times\tilde{\mathbf{A}} - \mathbf{M}) - \sigma\mathbf{v}\times(\nabla\times\tilde{\mathbf{A}}) = \mathbf{J}^{\mathrm{e}} + j\omega\mathbf{P}$$

Working with $\tilde{\mathbf{A}}$ is often the best option when it is possible to specify all source currents as external currents $\mathbf{J}^{\mathrm{e}}$ or as surface currents on boundaries.

Theory of Magnetic Fields

Simulation of magnetic fields is of interest when studying magnets, motors, transformers, and conductors carrying static or alternating currents.

The Magnetic Fields Interface is used for 3D, 2D in-plane, and 2D axisymmetric models. Unless you have a license for the AC/DC Module, only 2D modeling involving out-of-plane currents and axisymmetric modeling involving azimuthal currents are supported.

For a deeper theoretical background to the magnetic vector potential used, see the section starting with Maxwell's Equations.

Magnetostatics Equation

To derive the magnetostatic equation, start with Ampère's law for static cases $\nabla\times\mathbf{H}=\mathbf{J}$. The current is

$$\mathbf{J} = \sigma\mathbf{v}\times\mathbf{B}+\mathbf{J}^{\mathrm{e}}$$

where $\mathbf{J}^{\mathrm{e}}$ is an externally generated current density, and $\mathbf{v}$ is the velocity of the conductor.

Using the definitions of magnetic potential, $\mathbf{B}=\nabla\times\mathbf{A}$ and the constitutive relationship, $\mathbf{B}=\mu_0(\mathbf{H}+\mathbf{M})$, rewrite Ampère's law as

$$\nabla\times(\mu_0^{-1}\nabla\times\mathbf{A}-\mathbf{M})-\sigma\mathbf{v}\times(\nabla\times\mathbf{A}) = \mathbf{J}^{\mathrm{e}}$$

which is the equation used in magnetostatics.

Frequency Domain Equation

To derive the time harmonic equation this physics interface solves, start with Ampère's law including displacement currents (then called Maxwell-Ampère's law) as these do not involve any extra computational cost in the frequency domain,

$$\nabla\times\mathbf{H} = \mathbf{J}+\frac{\partial\mathbf{D}}{\partial t} = \sigma\mathbf{E}+\sigma\mathbf{v}\times\mathbf{B}+\mathbf{J}^{e}+\frac{\partial\mathbf{D}}{\partial t}$$

Now assume time-harmonic fields and use the definitions of the fields,

$$\begin{aligned}\mathbf{B} &= \nabla\times\mathbf{A}\\ \mathbf{E} &= -j\omega\mathbf{A}\end{aligned}$$

and combine them with the constitutive relationships $\mathbf{B}=\mu_0(\mathbf{H}+\mathbf{M})$ and $\mathbf{D}=\varepsilon_0\mathbf{E}$ to rewrite Ampère's law as

$$(j\omega\sigma-\omega^2\varepsilon_0)\mathbf{A}+\nabla\times(\mu_0^{-1}\nabla\times\mathbf{A}-\mathbf{M})-\sigma\mathbf{v}\times(\nabla\times\mathbf{A}) = \mathbf{J}^{\mathrm{e}}$$

Transient Equation

The transient equation this physics interface solves is Ampère's law, here illustrated with the constitutive relation $\mathbf{B}=\mu_0(\mathbf{H}+\mathbf{M})$.

$$\sigma\frac{\partial\mathbf{A}}{\partial t} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma\mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}_\mathrm{e}$$

The Electrostatics Interface

The **Electrostatics (es)** interface (⚡), found under the **AC/DC** branch (⚡) when adding a physics interface, is used to compute the electric field, the electric displacement field and potential distributions in dielectrics under conditions where the electric charge distribution is explicitly prescribed. The formulation is stationary but for use together with other physics, also eigenfrequency, frequency-domain, small-signal analysis and time-domain modeling are supported in all space dimensions.

The physics interface solves Gauss' Law for the electric field using the scalar electric potential as the dependent variable.

Charge Conservation is the main node, which adds the equation for the electric potential and has a settings window for defining the constitutive relation for the electric displacement field and its associated properties such as the relative permittivity.

When this physics interface is added, these default nodes are also added to the **Model Builder**—**Charge Conservation**, **Zero Charge** (the default boundary condition), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and space charges. You can also right-click **Electrostatics** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `es`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the electric potential and the equations that describe the potential field for dielectrics. To choose specific domains, select **Manual** from the **Selection** list.

THICKNESS

For 1D components, enter a default value for the **Cross-section area** A (SI unit: m^2). The default value of 1 is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models. See also Change Cross-Section.

Enter a default value for the **Out-of-plane thickness** d (SI unit: m). The default value of **1** is typically not representative for a thin dielectric medium, for example. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models. See also Change Thickness (Out-of-Plane).

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select an element order for the **Electric potential**—**Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or (in 2D only) **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

DEPENDENT VARIABLES

The dependent variable is the **Electric potential** V. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another electric potential field in the model, the physics interfaces shares degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface
- Theory of Electrostatics

Electric Sensor: model library path **COMSOL_Multiphysics/Electromagnetics/electric_sensor**

Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface

The Electrostatics Interface has these domain, boundary, edge, point, and pair nodes available.

ABOUT THE BOUNDARY CONDITIONS

The relevant physics interface condition at interfaces between different media is

$$\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

In the absence of surface charges, this condition is fulfilled by the natural boundary condition

$$\mathbf{n} \cdot [(\varepsilon_0 \nabla V - \mathbf{P})_1 - (\varepsilon_0 \nabla V - \mathbf{P})_2] = -\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

AVAILABLE NODES

These nodes, listed in alphabetical order, are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users). Also see Table 10-2 for a list of interior and exterior boundary conditions, including edge, point, and pair availability.

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

- Change Cross-Section
- Change Thickness (Out-of-Plane)
- Charge Conservation
- Electric Displacement Field
- Electric Potential
- External Surface Charge Accumulation
- Ground
- Initial Values
- Line Charge
- Line Charge (on Axis)
- Line Charge (Out-of-Plane)
- Periodic Condition
- Point Charge
- Point Charge (on Axis)
- Space Charge Density
- Surface Charge Density
- Thin Low Permittivity Gap
- Zero Charge (the default boundary condition)

Table 10-2 lists the interior and exterior boundaries available with this physics interface. It also includes edge, point, and pair availability.

TABLE 10-2: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTROSTATICS INTERFACE

NODE	INTERIOR	EXTERIOR	ALSO AVAILABLE FOR
Change Cross-Section	x	x	pairs
Change Thickness (Out-of-Plane)	x	x	pairs
Electric Displacement Field	x	x	pairs
Electric Potential	x	x	edges, points, and pairs
External Surface Charge Accumulation		x	pairs
Ground	x	x	edges, points, and pairs
Periodic Condition		x	not applicable
Surface Charge Density	x	x	pairs
Thin Low Permittivity Gap	x		not applicable
Zero Charge (the default)	x	x	pairs

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only. There are also **Line Charge (on Axis)** and **Point Charge (on Axis)** available.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

Charge Conservation

The **Charge Conservation** node adds the equations for charge conservation according to Gauss' law for the electric displacement field. It provides an interface for defining the constitutive relation and its associated properties such as the relative permittivity.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define the electric potential and the equation based on Gauss' law that describes the potential field or select **All domains** as required.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty.

MATERIAL TYPE

The **Material type** setting decides how materials behave and how material properties are interpreted when the mesh is deformed. Select **Solid** for materials whose properties change as functions of material strain, material orientation and other variables evaluated in a material reference configuration (material frame). Select **Non-solid** for materials whose properties are defined only as functions of the current local state at each point in the spatial frame, and for which no unique material reference configuration can be defined. Select **From material** to pick up the corresponding setting from the domain material on each domain.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

ELECTRIC FIELD

Select a **Constitutive relation** to describe the macroscopic properties of the medium (relating the electric displacement $\mathbf{D}$ with the electric field $\mathbf{E}$) and the applicable material properties, such as the relative permittivity.

Select:

- **Relative permittivity** (the default) to use the constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E}$. Then the default is to take the **Relative permittivity** ε_r (dimensionless) values **From material**. If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix. The default is 1.
- **Polarization** to use the constitutive relation $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$. Then enter the components based on space dimension for the **Polarization** vector $\mathbf{P}$ (SI unit: C/m^2). The defaults are 0 C/m^2.
- **Remanent electric displacement** to use constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E} + \mathbf{D}_r$, where $\mathbf{D}_r$ is the remanent displacement (the displacement when no electric field is present). Then the default is to take the **Relative permittivity** ε_r (dimensionless) values **From material**. If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix. Then enter the components based on space dimension for the **Remanent electric displacement** $\mathbf{D}_r$ (SI unit: C/m^2). The defaults are 0 C/m^2.

Initial Values

The **Initial Values** node adds an initial value for the electric potential V that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the initial value of the **Electric potential** V (SI unit: V). The default value is 0 V.

Space Charge Density

The **Space Charge Density** node adds a space charge density ρ, which appears on the right-hand side of the equation that the physics interface defines.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

SPACE CHARGE DENSITY

Enter a value or expression for the **Space charge density** ρ_q (SI unit: $\mathrm{C/m^3}$). The default is 0 $\mathrm{C/m^3}$.

Zero Charge

The **Zero Charge** node adds the condition that there is zero charge on the boundary so that $\mathbf{n} \cdot \mathbf{D} = 0$. This boundary condition is also applicable at symmetry boundaries where the potential is known to be symmetric with respect to the boundary. This is the default boundary condition at exterior boundaries. At interior boundaries, it means that no displacement field can penetrate the boundary and that the electric potential is discontinuous across the boundary.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Ground

The **Ground** node is the default boundary condition and implements ground (zero potential) as the boundary condition $V = 0$.

Ground means that there is a zero potential on the boundary. This boundary condition is also applicable at symmetry boundaries where the potential is known to be antisymmetric with respect to the boundary.

For some physics interfaces, also select additional **Ground** nodes from the **Edges** (3D models) or **Points** (2D and 3D models) submenus. For 2D axisymmetric models, it can be applied on the Symmetry axis.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (boundaries, edges, or points) to define.

> Beware that constraining the potential on edges or points in 3D or on points in 2D usually yields a current outflow that is mesh dependent.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Electric Potential

The **Electric Potential** node provides an electric potential V_0 as the boundary condition $V = V_0$.

Because the electric potential is being solved for in the physics interface, the value of the potential is typically defined at some part of the geometry. For some physics interfaces, also select additional **Electric Potential** nodes from the **Edges** (3D models) or **Points** (2D and 3D models) submenus. For 2D axisymmetric models, it can be applied on the symmetry axis.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entities (boundaries, edges, or points) to define.

> Beware that constraining the potential on edges or points in 3D or on points in 2D usually yields a current outflow that is mesh dependent.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

ELECTRIC POTENTIAL

Enter the value or expression for the **Electric potential** V_0 (SI unit: V). The default is 0 V.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Surface Charge Density

The **Surface Charge Density** node provides the following surface-charge boundary condition for exterior boundaries (left) and interior boundaries (right):

$$-\mathbf{n}\cdot\mathbf{D} = \rho_\mathrm{s}, \qquad \mathbf{n}\cdot(\mathbf{D}_1-\mathbf{D}_2) = \rho_\mathrm{s}$$

Specify the surface charge density ρ_s at an outer boundary or at an interior boundary between two nonconducting media. Also right-click to add a **Harmonic Perturbation** subnode.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

SURFACE CHARGE DENSITY

Enter the value or expression for the **Surface charge density** ρ_s (SI unit: C/m^2).

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

External Surface Charge Accumulation

The **External Surface Charge Accumulation** node implements the boundary condition

$$-\mathbf{n}\cdot\mathbf{D} = \rho_\mathrm{s}$$

where ρ_s is the solution of the following distributed ODE on the boundary:

$$\frac{d\rho_\mathrm{s}}{dt} = \mathbf{n}\cdot\mathbf{J}_i+\mathbf{n}\cdot\mathbf{J}_e$$

where $\mathbf{n}\cdot\mathbf{J}_i$ is the normal component of the total ion current density on the wall and $\mathbf{n}\cdot\mathbf{J}_e$ is the normal component of the total electron current density on the wall, which are feature inputs.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

EXTERNAL SURFACE CHARGE ACCUMULATION

Enter values or expressions for the **Normal ion current density** $\mathbf{n}\cdot\mathbf{J}_i$ (SI unit: A/m^2) and the **Normal electron current density** $\mathbf{n}\cdot\mathbf{J}_\mathbf{e}$ (SI unit: A/m^2). The defaults are 0 A/m^2 for both.

Electric Displacement Field

The **Electric Displacement Field** node adds the following electric-displacement boundary condition:

$$\mathbf{n}\cdot\mathbf{D} = \mathbf{n}\cdot\mathbf{D}_0$$

It specifies the normal component of the electric displacement field at a boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

ELECTRIC DISPLACEMENT FIELD

Enter the coordinates of the **Boundary electric displacement field** $\mathbf{D}_0$ (SI unit: $\mathrm{C/m^2}$). the defaults are $0\ \mathrm{C/m^2}$.

Periodic Condition

The **Periodic Condition** node defines periodicity or antiperiodicity between two boundaries. If required, activate periodic conditions on more than two boundaries, in which case the Periodic Condition tries to identify two separate surfaces that can each consist of several connected boundaries.

For more complex geometries it might be necessary to use the **Destination Selection** subnode. With this subnode the boundaries which constitute the source and destination surfaces can be manually specified. To add the subnode, right-click the **Periodic Condition** node and select **Destination Selection**.

When this feature is used in conjunction with a Sector Symmetry feature on connected boundaries, the same periodic condition feature cannot be used on both sides of where the sector symmetry boundaries connect with the periodic boundaries. At least two periodic condition features are required for the model to compute correctly.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

When using non conforming meshes on the source and destination of a periodic boundary pair, for numerical stability a finer mesh should be applied on the destination side. Use conforming meshes if possible.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or

Individual dependent variables to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

Thin Low Permittivity Gap

Use the **Thin Low Permittivity Gap** node

$$\mathbf{n}\cdot\mathbf{D}_1 = \frac{\varepsilon_0\varepsilon_\mathrm{r}}{d_s}(V_1 - V_2)$$

$$\mathbf{n}\cdot\mathbf{D}_2 = \frac{\varepsilon_0\varepsilon_\mathrm{r}}{d_s}(V_2 - V_1)$$

to model a thin gap of a material with a small permittivity compared to the adjacent domains. The layer has the thickness d_s and the relative permittivity ε_r. The indices 1 and 2 refer to the two sides of the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

THIN LOW PERMITTIVITY GAP

The default is to take the **Relative permittivity** ε_r (dimensionless) values **From material**. Select **User defined** to enter a different value or expression. Enter a **Surface thickness** d_s (SI unit: m). The default is 5 mm.

Line Charge

For 3D models, use the **Line Charge** node to specify line charges along the edges of a geometry. Also right-click to add a **Harmonic Perturbation** subnode.

EDGE SELECTION

From the **Selection** list, choose the edges to define.

Beware that constraining the potential on edges usually yields a current outflow that is mesh dependent.

LINE CHARGE

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length and the default is 0 C/m.

- Line Charge (on Axis)
- Line Charge (Out-of-Plane)
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Line Charge (on Axis)

For 2D axisymmetric models, use the **Line Charge (on Axis)** node to specify line charges along the symmetry axis. Also right-click to add a **Harmonic Perturbation** subnode.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

LINE CHARGE (ON AXIS)

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length and the default is 0 C/m.

- Line Charge
- Line Charge (Out-of-Plane)
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Line Charge (Out-of-Plane)

For 2D and 2D axisymmetric models, points are selected and this is the same as a line out-of-plane. Also right-click to add a **Harmonic Perturbation** subnode.

Use the **Line Charge (Out-of-Plane)** node to specify line charges along the points of a geometry for 2D and 2D axisymmetric models.

POINT SELECTION

From the **Selection** list, choose the points to define.

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

LINE CHARGE (OUT-OF-PLANE)

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length and the default is 0 C/m.

- Line Charge
- Line Charge (on Axis)
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Point Charge

The **Point Charge** node adds a point source to 3D models. The point charge represents an electric displacement field flowing out of the point. Also right-click to add a **Harmonic Perturbation** subnode.

POINT SELECTION

From the **Selection** list, choose the points to define.

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CHARGE

Enter a value or expression to apply a **Point charge** Q_P (SI unit: C) to points. This source represents an electric displacement field flowing out of the point. The default is 0 C.

- Point Charge (on Axis)
- Line Charge (Out-of-Plane)
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Point Charge (on Axis)

The **Point Charge (on Axis)** node adds a point source to 2D axisymmetric models. The point charge represents an electric displacement field flowing out of the point. Also right-click to add a **Harmonic Perturbation** subnode.

POINT SELECTION

From the **Selection** list, choose the points to define.

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CHARGE (ON AXIS)

Enter a value or expression to apply a **Point charge** Q_P (SI unit: C) to points on axis. This source represents an electric displacement field flowing out of the point. The default is 0 C.

- Point Charge
- Line Charge (Out-of-Plane)
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Change Cross-Section

This node is available with 1D models. This setting overrides the global **Thickness** setting made in any physics interface that uses this feature.

Use the **Change Cross-Section** node to set the cross-section area for specific geometric entities.

DOMAIN OR BOUNDARY SELECTION

From the **Selection** list, choose the geometric entity (domains or boundaries) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CHANGE CROSS-SECTION

Enter a value or expression for the **Cross-section area** A (SI unit: m^2). The default value of 1 unit length is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models.

For 2D models, see Change Thickness (Out-of-Plane).

Change Thickness (Out-of-Plane)

This node is available for 2D models. This setting overrides the global **Thickness** setting made in any physics interface that uses this node.

Use the **Change Thickness (Out-of-Plane)** node to set the out-of-plane thickness for specific geometric entities.

DOMAIN OR BOUNDARY SELECTION

From the **Selection** list, choose the geometric entity (domains or boundaries) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CHANGE THICKNESS (OUT-OF-PLANE)

Enter a value or expression for the **Out-of-plane thickness** d (SI unit: m). The default value of 1 unit length is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models.

For 1D models, see Change Cross-Section.

The Electric Currents Interface

The **Electric Currents (ec)** interface ( ), found under the **AC/DC** branch ( ) when adding a physics interface, is used to compute electric field, current and potential distributions in conducting media under conditions where inductive effects are negligible; that is, when the skin depth is much larger than the studied device.

Depending on the licensed products, stationary, frequency-domain, small-signal analysis, and time-domain modeling are supported in all space dimensions. In the time and frequency domains, capacitive effects are also accounted for.

The physics interface solves a current conservation equation based on Ohm's law using the scalar electric potential as the dependent variable.

Current Conservation is the main node, which adds the equation for the electric potential and provides a settings window for defining the electrical conductivity as well as the constitutive relation for the electric displacement field and its associated material properties such as the relative permittivity.

When this physics interface is added, these default nodes are also added to the **Model Builder**—**Current Conservation**, **Electric Insulation** (the default boundary condition), and **Initial Values**. Then from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and current sources. You can also right-click **Electric Currents** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `ec`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the electric potential and the equations that describe the potential field for conductive media. To choose specific domains, select **Manual** from the **Selection** list.

THICKNESS (1D)/ OUT-OF-PLANE THICKNESS (2D)

For 1D components, enter a default value for the **Cross-section area** A (SI unit: m^2). The default value of 1 is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models. See also Change Cross-Section (described for the Electrostatics interface).

Enter a default value for the **Out-of-plane thickness** d (SI unit: m) (see Equation 10-1). The default value of 1 m is typically not representative for a thin dielectric medium, for example. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models. See also Change Thickness (Out-of-Plane) (described for the Electrostatics interface).

DEPENDENT VARIABLES

The dependent variable is the **Electric potential** V. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another electric potential field in the model, the physics interfaces share degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select an **Electric potential**—**Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface
- Theory of Electric Currents

Pacemaker Electrode: model library path **COMSOL_Multiphysics/Electromagnetics/pacemaker_electrode**

Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface

The Electric Currents Interface has these domain, boundary, edge, point, and pair nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. However, to add subnodes, right-click the parent node.

ABOUT THE BOUNDARY CONDITIONS

The exterior and interior boundary conditions listed in Table 10-2 are available. The relevant physics interface condition at interfaces between different media and interior boundaries is continuity; that is,

$$\mathbf{n}_2 \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

which is the natural boundary condition.

AVAILABLE NODES

These nodes are available for this physics interface and listed in alphabetical order. Also see Table 10-2 for a list of interior and exterior boundary conditions, including edge, point, and pair availability.

- Boundary Current Source
- Contact Impedance
- Current Conservation
- Current Source
- Distributed Impedance
- Electric Insulation
- External Current Density
- Initial Values
- Line Current Source
- Line Current Source (on Axis)
- Normal Current Density
- Point Current Source
- Sector Symmetry

These nodes are described for the Electrostatics interface:

- Change Cross-Section
- Change Thickness (Out-of-Plane)
- Electric Potential
- Ground
- Periodic Condition

You can right-click many of the nodes to add the Harmonic Perturbation subnode. For more information see Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Table 10-2 lists the interior and exterior boundaries available with this physics interface. It also includes edge, point, and pair availability.

TABLE 10-3: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTRIC CURRENTS INTERFACE

NODE	INTERIOR	EXTERIOR	ALSO AVAILABLE FOR
Boundary Current Source	x		pairs
Contact Impedance	x		pairs
Distributed Impedance	x	x	not applicable
Electric Insulation	x	x	pairs
Electric Potential	x	x	edges, points, and pairs
Ground	x	x	edges, points, and pairs
Normal Current Density		x	not applicable
Periodic Condition		x	not applicable

Current Conservation

The **Current Conservation** node adds the continuity equation for the electrical potential and provides an interface for defining the electric conductivity as well as the constitutive relation and the relative permittivity for the displacement current.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define the electric potential and the continuity equation that describes the potential field or select **All domains** as required.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty. If a linear temperature relation is added for the conductivity, then the source for the temperature T can be defined. From the **Temperature** list, select an existing temperature variable (from another physics interface) if available, or select **User defined** to define a value or expression for the temperature (SI unit: K) in the field that appears underneath the list.

MATERIAL TYPE

The **Material type** setting decides how materials behave and how material properties are interpreted when the mesh is deformed. Select **Solid** for materials whose properties change as functions of material strain, material orientation and other variables evaluated in a material reference configuration (material frame). Select **Non-solid** for materials whose properties are defined only as functions of the current local state at each point in the spatial frame, and for which no unique material reference configuration can be defined. Select **From material** to pick up the corresponding setting from the domain material on each domain.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

CONDUCTION CURRENT

By default, the **Electrical conductivity** σ (SI unit: S/m) for the media is defined **From material**. Or select **User defined** or **Linearized resistivity**.

User Defined

If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** depending on the characteristics of the electrical conductivity, and then enter values or expressions for the electrical conductivity σ in the field or matrix. The default is 0 S/m.

If another type of temperature dependence is used other than a linear temperature relation, enter any expression for the conductivity as a function of temperature.

Linearized Resistivity

Select **Linearized resistivity** for a temperature-dependent conductivity (this occurs in, for example, Joule heating, and is also called resistive heating). The equation describing the conductivity:

$$\sigma = \frac{1}{\rho_0(1+\alpha(T-T_0))}$$

where ρ_0 is the resistivity at the reference temperature T_0, and α is the temperature coefficient of resistance, which describes how the resistivity varies with temperature.

The default **Reference resistivity** ρ_0 (SI unit: Ω·m), **Reference temperature** T_{ref} (SI unit: K), and **Resistivity temperature coefficient** α (SI unit: 1/K) are taken **From material**, which means that the values are taken from the domain (or boundary) material. T is the current temperature, which can be a value that is specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field appears in the **Model Inputs** section.

To specify other values for any of these properties, select **User defined** from the list and then enter a value or expression for each. The default values are:

- 1 Ω·m for the Reference resistivity
- 273.15 K for the Reference temperature, and
- 0 1/K for the Resistivity temperature coefficient

ELECTRIC FIELD

See **Electric Field** as described for the Charge Conservation node for the Electrostatics interface.

Initial Values

The **Initial Values** node adds an initial value for the electric potential that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. If more than one set of initial values is required, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and current sources. Add more **Initial Values** nodes from the **Physics** toolbar.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains.

INITIAL VALUES

Enter a value or expression for the initial value of the **Electric potential** V (SI unit: V). The default value is 0 V.

External Current Density

The **External Current Density** node adds an externally generated current density $\mathbf{J}_\mathrm{e}$, which appears in Ohm's law

$$\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_\mathrm{e}$$

and in the equation that the physics interface defines. Also right-click to add a **Harmonic Perturbation** subnode.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

EXTERNAL CURRENT DENSITY

Based on space dimension, enter the coordinates (**x**, **y**, and **z** for 3D models for example) of the **External current density** $\mathbf{J}_e$ (SI unit: A/m^2). The defaults are 0 A/m^2.

Current Source

The **Current Source** node adds a distributed current source Q_j in the equation that the physics interface defines. Use this node with caution as it can violate the current conservation law that is inherent in Maxwell-Ampère's law. Also right-click to add a **Harmonic Perturbation** subnode.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

CURRENT SOURCE

Enter a value or expression for the **Current source** Q_j (SI unit: A/m^3). The default is 0 A/m^3.

Electric Insulation

The **Electric Insulation** node, which is the default boundary condition, adds electric insulation as the boundary condition:

$$\mathbf{n} \cdot \mathbf{J} = 0$$

This boundary condition means that no electric current flows into the boundary. At interior boundaries, it means that no current can flow through the boundary and that the electric potential is discontinuous across the boundary. It is also applicable at symmetric boundaries where the potential is known to be symmetric with respect to the boundary.

Electric insulation as the default boundary condition is not applicable to interior boundaries.

> To add electric insulation to an interior boundary, add an **Electric Insulation** node in addition to the one that represents the default boundary condition.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Boundary Current Source

The **Boundary Current Source** node adds a current source Q_j on the boundary.

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = Q_j$$

It is applicable to interior boundaries that represent either a source or a sink of current. Also right-click to add a **Harmonic Perturbation** subnode.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

BOUNDARY CURRENT SOURCE

Enter a value or expression for the **Boundary current source** Q_j (SI unit: A/m^2). The default is 0 A/m^2.

Normal Current Density

The **Normal Current Density** node is applicable to exterior boundaries that represent either a source or a sink of current. It provides a condition for specifying the normal current density as an inward or outward current flow:

$$-\mathbf{n}\cdot\mathbf{J} = J_n$$

Or, alternatively, as a current density $\mathbf{J}_0$:

$$\mathbf{n}\cdot\mathbf{J} = \mathbf{n}\cdot\mathbf{J}_0$$

The normal current density is positive when the current flows inward in the domain. Also right-click to add a **Harmonic Perturbation** subnode.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a current flow as the boundary condition using the normal current density.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

NORMAL CURRENT DENSITY

Select a **Type**—**Inward current density** (the default) or **Current density**.

- If **Inward current density** is selected, enter a value or expression for the **Normal current density** J_n (SI unit: A/m^2). Use a positive value for an inward current flow or a negative value for an outward current flow. The default is 0 A/m^2.
- If **Current density** is selected, enter values or expressions for the components of the **Current density** $\mathbf{J}_0$ (SI unit: A/m^2). The defaults are 0 A/m^2.

Distributed Impedance

The **Distributed Impedance** node adds a distributed impedance boundary condition to a model. Also right-click to add a **Harmonic Perturbation** subnode.

> Use this boundary condition to model a thin sheet of a resistive material, connected to a reference potential V_{ref}.

The layer impedance can be specified either with the bulk material conductivity σ_s, the relative permittivity ε_r and the layer thickness d_s, or directly with the surface resistance ρ_s and capacitance C_s. Assuming DC currents, the equation is:

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = \frac{\sigma_s}{d_s}(V-V_{\mathrm{ref}})$$

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = \frac{1}{\rho_s}(V-V_{\mathrm{ref}})$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

DISTRIBUTED IMPEDANCE

Enter the reference potential V_{ref} (SI unit: V). The default is 0 V.

Select a potentially complex valued **Layer specification**—**Thin layer** (the default) or **Surface impedance**.

- If **Thin layer** is selected, enter values or expressions for the:
  - **Surface thickness** d_{s} (SI unit: m). The default is $5\cdot10^{-3}$ m (5 mm).
  - **Electrical conductivity** σ (SI unit: S/m) and **Relative permittivity** ε_r (dimensionless). The defaults take values **From material**. Select **User defined** to enter different values or expressions. The default electrical conductivity is $1\cdot10^{-2}$ S/m and the default relative permittivity is 1.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: Ω·m^2) and the **Surface capacitance** C_{s} (SI unit: F/m^2). The default surface impedance is $1\cdot10^{-8}$ Ω·m^2 and the default surface capacitance is 0 F/m^2.

Contact Impedance

Use the **Contact Impedance** node on interior boundaries to model a thin layer of resistive material. It can also be added as a pair using a **Pair Contact Impedance** node.

$$\mathbf{n}\cdot\mathbf{J}_1 = \frac{\sigma}{d_s}(V_1 - V_2)$$

$$\mathbf{n}\cdot\mathbf{J}_2 = \frac{\sigma}{d_s}(V_2 - V_1)$$

$$\mathbf{n}\cdot\mathbf{J}_1 = \frac{1}{\rho_s}(V_1 - V_2)$$

$$\mathbf{n}\cdot\mathbf{J}_2 = \frac{1}{\rho_s}(V_2 - V_1)$$

The layer impedance can be specified either with the bulk material conductivity σ_s, the relative permittivity ε_r and the layer thickness d_s, or directly with the surface resistance ρ_s and capacitance C_s. The indices 1 and 2 refer to the two sides of the boundary.

These parameters work the same as with Distributed Impedance.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONTACT IMPEDANCE

Select a potentially complex valued **Layer specification**—**Thin layer** (the default) or **Surface impedance**.

- If **Thin layer** is selected, enter values or expressions for the:
  - **Surface thickness** d_s (SI unit: m). The default is $5{\cdot}10^{-3}$ m (5 mm).
  - **Electrical conductivity** σ (SI unit: S/m) and **Relative permittivity** ε_r (dimensionless). The defaults take values **From material**. Select **User defined** to enter different values or expressions. The default electrical conductivity is $1{\cdot}10^{-2}$ S/m and the default relative permittivity is 1.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: $\Omega{\cdot}\text{m}^2$) and the **Surface capacitance** C_s (SI unit: F/m^2). The default surface impedance is $1{\cdot}10^{-8}$ $\Omega{\cdot}\text{m}^2$ and the default surface capacitance is 0 F/m^2.

Thin-Film Resistance: model library path
COMSOL_Multiphysics/Electromagnetics/thin_film_resistance

Sector Symmetry

Select **Sector Symmetry** at interfaces between rotating objects where sector symmetry is used. It is only available for pairs. A default subnode is added. Right-click to select additional features from the **Fallback Features** submenu.

In 2D, this feature assumes rotation around the origin.

This feature is always used in conjunction with a Periodic Condition on adjacent radial sector boundaries. Note that the same periodic condition feature cannot be used on both sides of where the sector symmetry boundaries connect with the periodic boundaries. At least two periodic condition features are required for the model to compute correctly.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries from an existing identity pair. This pair first has to be created.

PAIR SELECTION

Choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

When using non conforming meshes on the source and destination of a pair, for numerical stability a finer mesh should be applied on the destination side for any pair with a condition that imposes a coupling or a constraint across the pair. The sector symmetry feature falls into this category.

SECTOR SETTINGS

Enter the **Number of sectors (<50)** n_{sect}. The default is 2.

Select a **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

Based on space dimension, enter values or expressions in the table for the **Axis of rotation** $\mathbf{a}_{\mathrm{rot}}$.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

Line Current Source

The **Line Current Source** node adds a line source to edges in 3D models and to points in 2D and 2D axisymmetric models. The line source represents electric current per unit length.

EDGE OR POINT SELECTION

From the **Selection** list, choose the edges or points to define.

Beware that constraining the potential on edges or points usually yields a current outflow that is mesh dependent.

LINE CURRENT SOURCE

Enter a value or expression to apply a **Line current source** Q_j (SI unit: A/m). This source represents electric current per unit length. The default is 0 A/m.

Line Current Source (on Axis) for 2D axisymmetric models

Line Current Source (on Axis)

The **Line Current Source (on Axis)** node adds a line source to boundaries in 2D axisymmetric models. The line source represents electric current per unit length.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

LINE CURRENT SOURCE (ON AXIS)

Enter a value or expression to apply a **Line current source** Q_j (SI unit: A/m) to boundaries. This source represents electric current per unit length.

Line Current Source

Point Current Source

The **Point Current Source** node adds a point source and represents an electric current flowing out of the point. Add point sources to 3D models from the **Points** menu. Also right-click to add a **Harmonic Perturbation** subnode.

POINT SELECTION

From the **Selection** list, choose the points to define.

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CURRENT SOURCE

Enter a value or expression to apply a **Point current source** Q_j (SI unit: A) to points. This source represents an electric current flowing out of the point.

- Line Current Source for 2D models
- Point Current Source (on Axis) for 2D axisymmetric models

Point Current Source (on Axis)

The **Point Current Source (on Axis)** node adds a point source and represents an electric current flowing out of the point in 2D axisymmetric models.

POINT SELECTION

From the **Selection** list, choose the points to define.

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CURRENT SOURCE

Enter a value or expression to apply a **Point current source** Q_j (SI unit: A) to points. This source represents an electric current flowing out of the point.

- Point Current Source for 3D models
- Line Current Source for 2D models

The Magnetic Fields Interface

The **Magnetic Fields (mf)** interface ( ), found under the **AC/DC** branch ( ) when adding a physics interface, is used to compute magnetic field and induced current distributions in and around coils, conductors and magnets. Depending on the licensed products, stationary, frequency-domain, small-signal analysis and time-domain modeling are supported in 2D and 3D. Note that the frequency and time domain formulations become ill-posed when approaching the static limit. One may extend the useful frequency range downward by adding a low conductivity.

The physics interface solves Maxwell's equations formulated using the magnetic vector potential and, optionally for coils, the scalar electric potential as the dependent variables.

The main node is **Ampère's Law**, which adds the equation for the magnetic vector potential and provides an interface for defining the constitutive relations and its associated properties such as the relative permeability.

When this physics interface is added, these default nodes are also added to the **Model Builder**— **Magnetic Fields**, **Ampère's Law**, **Magnetic Insulation** (the default boundary condition), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, boundary conditions and external currents. You can also right-click **Magnetic Fields** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `mf`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the magnetic vector potential and the equations that describe the potential field for magnetic fields. To choose specific domains, select **Manual** from the **Selection** list.

BACKGROUND FIELD

This section allows the specification of a background magnetic vector potential (that generates a background magnetic flux density). The only option to **Solve for** is **Full field**.

COMPONENTS

The current vector has the same direction as the magnetic vector potential. This setting also controls the direction in which applied and induced currents can flow in the model. The default option is to solve for the out-of-plane component only. Therefore, the only **Components** option is **Out-of-plane vector potential**.

THICKNESS

For 2D models, enter a value or expression for the global **Out-of-plane thickness** d (SI unit: m). The default of 1 m is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models.

Use the Change Thickness (Out-of-Plane) node (described for the Electrostatics interface) to define specific geometric entities (for example, domains) instead of a global setting for the thickness.

DEPENDENT VARIABLES

The dependent variable is the **Magnetic vector potential** A. You can change both its field name and the individual component variable names. If the new field name coincides with the name of another magnetic vector potential field in the model, the physics interfaces share degrees of freedom and component names. The new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model, except for fields of the same type sharing a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select a **Magnetic vector potential**—**Quadratic** (the default), **Linear**, or **Cubic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

- Show More Physics Options
- Domain, Boundary, Point, and Pair Nodes for the Magnetic Fields Interface
- Theory of Magnetic Fields

Quadrupole Lens: model library path **COMSOL_Multiphysics/Electromagnetics/quadrupole**

Domain, Boundary, Point, and Pair Nodes for the Magnetic Fields Interface

The Magnetic Fields Interface has these domain, boundary, point, and pair nodes available, which are listed in alphabetical order.

About the Boundary Conditions

With no surface currents present the physics interface conditions

$$\mathbf{n}_2 \times (\mathbf{A}_1 - \mathbf{A}_2) = \mathbf{0}$$

$$\mathbf{n}_2 \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{0}$$

need to be fulfilled. Because **A** is being solved for, the tangential component of the magnetic potential is always continuous, and thus the first condition is automatically fulfilled. The second condition is equivalent to the natural boundary condition and is hence also fulfilled unless surface currents are explicitly introduced.

Table 10-2 lists the interior and exterior boundaries available with this physics interface.

TABLE 10-4: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS FOR THE MAGNETIC FIELDS INTERFACE

NODE	INTERIOR	EXTERIOR
Change Thickness (Out-of-Plane)	x	x
Magnetic Field	x	x
Magnetic Insulation	x	x
Magnetic Potential	x	x
Perfect Magnetic Conductor	x	x
Periodic Condition		x
Surface Current	x	x

Available Nodes

These nodes, listed in alphabetical order, are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. However, to add subnodes, right-click the parent node.

Also see Table 10-2 for a list of interior and exterior boundary conditions.

- Ampère’s Law
- Change Thickness (Out-of-Plane) (described for the Electrostatics interface)External Current Density
- Initial Values
- Line Current (Out-of-Plane)
- Magnetic Field
- Magnetic Insulation (the default boundary condition)
- Magnetic Potential
- Perfect Magnetic Conductor
- Surface Current
- Velocity (Lorentz Term)

For 2D axisymmetric models, COMSOL takes the axial symmetry boundaries (at $r = 0$) into account and adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Infinite Element Domains and Perfectly Matched Layers

Ampère's Law

The **Ampère's Law** node adds Ampère's law for the magnetic field and provides an interface for defining the constitutive relation and its associated properties as well as electric properties.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node and cannot be edited; that is, the selection is automatically selected and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose the domains to define the magnetic vector potential and the equation based on Ampère's law that defines the potential or select **All domains** as required.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty. If a linear temperature relation is added for the conductivity, then define the source for the temperature T. From the **Temperature** list, select an existing temperature variable (from another physics interface) if available, or select **User defined** to define a value or expression for the temperature (SI unit: K) in the field that appears underneath the list.

MATERIAL TYPE

The **Material type** setting decides how materials behave and how material properties are interpreted when the mesh is deformed. Select **Solid** for materials whose properties change as functions of material strain, material orientation and other variables evaluated in a material reference configuration (material frame). Select **Non-solid** for materials whose properties are defined only as functions of the current local state at each point in the spatial frame, and for which no unique material reference configuration can be defined. Select **From material** to pick up the corresponding setting from the domain material on each domain.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

CONDUCTION CURRENT

This section is described for the Current Conservation feature.

ELECTRIC FIELD

The default **Relative permittivity** ε_r (dimensionless) for the media is used **From material** and defined on the shell domain. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the permittivity and then enter values or expressions in the field or matrix.

MAGNETIC FIELD

Specify the constitutive relation that describes the macroscopic properties of the medium (relating the magnetic flux density $\mathbf{B}$ and the magnetic field $\mathbf{H}$) and the applicable material properties, such as the relative permeability.

The equation for the selected constitutive relation displays under the list. For all options, the default uses values **From material**, or select **User defined** to enter a different value or expression.

Select a **Constitutive relation**—**Relative permeability** (the default), **HB curve**, **Magnetic losses**, **Remanent flux density**, or **Magnetization**.

Relative Permeability

Select **Relative permeability** μ_r (dimensionless) to use the constitutive relation $\mathbf{B} = \mu_0\mu_r\mathbf{H}$. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix.

HB Curve

Select **HB curve** $|\mathbf{H}|$ (SI unit: A/m) to use a curve that relates magnetic flux density **B** and the magnetic field **H** as $|\mathbf{H}| = f(|\mathbf{B}|)$.

Magnetic Losses

Select **Magnetic losses** μ' and μ'' (dimensionless) to describe the relative permeability as a complex-valued quantity: $\mu_r = \mu' + i\mu''$, where μ' and μ'' are the real and imaginary parts, respectively.

Remanent Flux Density

Select **Remanent flux density** $\mathbf{B}_r$ (SI unit: T) to use the constitutive relation $\mathbf{B} = \mu_0\mu_r\mathbf{H} + \mathbf{B}_r$, where $\mathbf{B}_r$ is the remanent flux density (the flux density when no magnetic field is present).

- The default relative permeability μ_r (dimensionless) uses values **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the relative permeability and enter another value or expression in the field or matrix.
- Enter **x** and **y** components for the **Remanent flux density** $\mathbf{B}_r$.

Magnetization

Select **Magnetization** **M** (SI unit: A/m) to use the constitutive relation $\mathbf{B} = \mu_0\mathbf{H} + \mu_0\mathbf{M}$. Enter **x** and **y** components.

Initial Values

The **Initial Values** node adds an initial value for the magnetic vector potential A that can serve as an initial value for a transient simulation or as an initial guess for a nonlinear solver.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter values or expressions for the **Magnetic vector potential** **A** (SI unit: Wb/m). The defaults are 0 Wb/m.

External Current Density

The **External Current Density** node adds an externally generated current density $\mathbf{J}_e$, which appears on the right-hand side of the equation that the Magnetic Fields interface defines. Also right-click to add a **Harmonic Perturbation** subnode. See Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis for more information.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

EXTERNAL CURRENT DENSITY

Enter a value or expression for each component of the **External current density** $\mathbf{J}_e$ (SI unit: A/m^2). The defaults are 0 A/m^2.

Velocity (Lorentz Term)

This node is only valid in 2D and 2D axisymmetry when only solving for the out-of-plane component of the magnetic vector potential.

> To use the velocity feature correctly requires deep physical insight. In situations when the moving domain is of bounded extent in the direction of the motion or material properties vary in this direction or it contains magnetic sources that also move, the Lorentz term must not be used.

The **Velocity (Lorentz term)** node adds velocity $\mathbf{v}$. The external current is equal to $\sigma\mathbf{v} \times \mathbf{B}$.

An operational definition of when it can be used is that the moving domain should only contain an induced magnetic source (magnetization plus eddy currents) that has to be stationary with respect to the motion. Thus, it cannot be used for modeling projectiles of finite length or projectiles containing magnets. It can be used to model conductive, homogeneous spinning disks (magnetic brakes), magnets over a moving infinite homogenous plane (maglev trains), a flow of homogeneous conducting fluid past a magnet (liquid metal pumps or Hall generators/thrusters, for example).

> If you are not sure how to proceed, contact the COMSOL Support Center: http://www.comsol.com/support.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

VELOCITY (LORENTZ TERM)

User defined is selected by default. Enter the components for the **Velocity** vector $\mathbf{v}$ (SI unit: m/s) or, if present, select any velocity field defined in the model.

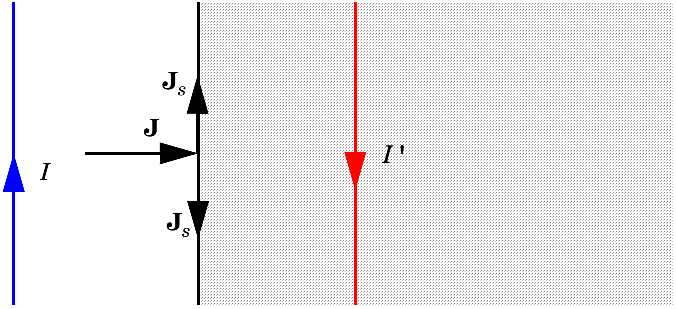
> For example, using the velocity field is useful when coupling to the velocity field of a fluid for a magnetohydrodynamic model.

Magnetic Insulation

The **Magnetic Insulation** node is the default boundary condition for the Magnetic Fields interface and adds a boundary condition that sets the tangential components of the magnetic potential to zero at the boundary $\mathbf{n} \times \mathbf{A} = 0$.

> Magnetic insulation is a special case of the magnetic potential boundary condition that sets the tangential component of the magnetic potential to zero.

It is used for the modeling of a lossless metallic surface, for example a ground plane or as a symmetry type boundary condition. It imposes symmetry for magnetic fields and “magnetic currents.” In the transient and time harmonic formulations it also imposes antisymmetry for electric fields and electric currents. It supports induced electric surface currents and thus any prescribed or induced electric currents (volume, surface, or edge currents) flowing into a perfect electric conductor boundary is automatically balanced by induced surface currents.



The magnetic insulation boundary condition is used on exterior and interior boundaries representing the surface of a lossless metallic conductor or (on exterior boundaries) representing a symmetry cut. The shaded (metallic) region is not part of the model but still carries effective mirror images of the sources. Note also that any current flowing into the boundary is perfectly balanced by induced surface currents. The tangential vector potential (and electric field) vanishes at the boundary.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, that is, **All boundaries** is automatically selected and this applies to all the *external* boundaries. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

> Weak constraints perform poorly when applied on vector elements. They should be used when the magnetic vector potential is discretized with Lagrange elements, for example when solving for out-of-plane component in a two-dimensional model.

Magnetic Field

The **Magnetic Field** node adds a boundary condition for specifying the tangential component of the magnetic field at the boundary:

$$\mathbf{n}\times\mathbf{H} = \mathbf{n}\times\mathbf{H}_0$$

Also right-click to add a **Harmonic Perturbation** subnode. See Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis for more information.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

MAGNETIC FIELD

Enter the value or expression for the **Magnetic Field** $\mathbf{H}_0$ (SI unit: A/m) vector coordinates. The defaults are 0 A/m.

Surface Current

The **Surface Current** node adds a boundary condition for a surface current density $\mathbf{J}_s$:

$$-\mathbf{n}\times\mathbf{H} = \mathbf{J}_s$$

$$\mathbf{n}\times(\mathbf{H}_1-\mathbf{H}_2) = \mathbf{J}_s$$

Also right-click to add a **Harmonic Perturbation** subnode. See Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis for more information.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

SURFACE CURRENT

Enter values or expressions for the **Surface current density** $\mathbf{J}_{s0}$ (SI unit A/m) coordinates. The defaults are 0 A/m.

Magnetic Potential

The **Magnetic Potential** node adds a boundary condition for the magnetic vector potential:

$$\mathbf{n}\times\mathbf{A} = \mathbf{n}\times\mathbf{A}_0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

MAGNETIC POTENTIAL

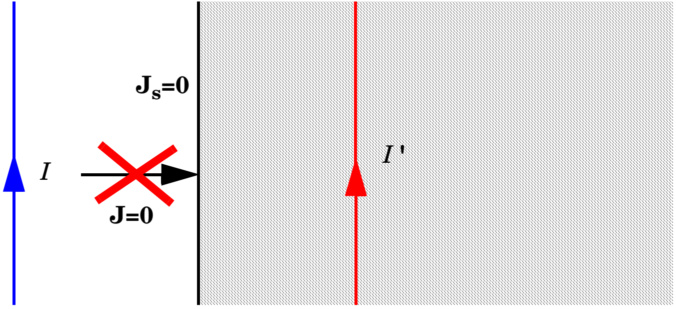
Enter a value or expression for the **Magnetic vector potential** $\mathbf{A}_0$ (SI unit: Wb/m) coordinates.

CONSTRAINT SETTINGS

These settings are the same as for Magnetic Insulation.

Perfect Magnetic Conductor

The **Perfect Magnetic Conductor** boundary condition $\mathbf{n} \times \mathbf{H} = 0$ is a special case of the surface current boundary condition that sets the tangential component of the magnetic field and thus also the surface current density to zero. On external boundaries, this can be interpreted as a “high surface impedance” boundary condition or used as a symmetry type boundary condition. It imposes symmetry for electric fields and electric currents. Electric currents (volume, surface, or edge currents) are not allowed to flow into a perfect magnetic conductor boundary as that would violate current conservation. On interior boundaries, the perfect magnetic conductor boundary condition literally sets the tangential magnetic field to zero which in addition to setting the surface current density to zero also makes the tangential magnetic vector potential (and in dynamics the tangential electric field) discontinuous.



The perfect magnetic conductor boundary condition is used on exterior boundaries representing the surface of a high impedance region or a symmetry cut. The shaded (high impedance) region is not part of the model but nevertheless carries effective mirror images of the sources. Note also that any electric current flowing into the boundary is forbidden as it cannot be balanced by induced electric surface currents. The tangential magnetic field vanishes at the boundary. On interior boundaries, the perfect magnetic conductor boundary condition literally sets the tangential magnetic field to zero which in addition to setting the surface current density to zero also makes the tangential magnetic vector potential (and in dynamics the tangential electric field) discontinuous.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Line Current (Out-of-Plane)

Use the **Line Current (Out-of-Plane)** node, selected from the **Points** menu, to specify a line current out of the modeling plane. In axially symmetric geometries this is the rotational (azimuthal) direction, in 2D geometries this is the z-direction.

POINT SELECTION

From the **Selection** list, choose the points to define.

LINE CURRENT (OUT-OF-PLANE)

Enter a value or expression for the **Out-of-plane current** I_0 (SI unit: A). The default is 0 A.

11

Pressure Acoustics

This chapter describes how to use the Pressure Acoustics, Frequency Domain interface, found under the **Acoustics>Pressure Acoustics** branch (◎) when adding a physics interface, for modeling and simulation of acoustics and vibrations.

Fundamentals of Acoustics

There are certain difficulties that often arise when modeling acoustics, such as the rather severe requirements on the mesh resolution, the modeling of artificial boundaries, and the modeling of real-world damping materials. This section also includes a brief introduction to acoustics, gives some examples of standard acoustics problems, and provides a short introduction to the mathematical formulation of the governing equations.

Acoustics Explained

Acoustics is the physics of sound. Sound is the sensation, as detected by the ear, of very small rapid changes in the air pressure above and below a static value. This static value is the atmospheric pressure (about 100,000 pascals), which varies slowly. Associated with a sound pressure wave is a flow of energy—the intensity. Physically, sound in air is a longitudinal wave where the wave motion is in the direction of the movement of energy. The wave crests are the pressure maxima, while the troughs represent the pressure minima.

Sound results when the air is disturbed by some source. An example is a vibrating object, such as a speaker cone in a sound system. It is possible to see the movement of a bass speaker cone when it generates sound at a very low frequency. As the cone moves forward it compresses the air in front of it, causing an increase in air pressure. Then it moves back past its resting position and causes a reduction in air pressure. This process continues, radiating a wave of alternating high and low pressure propagating at the speed of sound.

The propagation of sound in solids happens through small-amplitude elastic oscillations of its shape. These elastic waves are transmitted to surrounding fluids as ordinary sound waves. The elastic sound waves in the solid are the counter part to the pressure waves or compressible waves propagating in the fluid.

Examples of Standard Acoustics Problems

Depending on the basic dependent variable used to model the acoustic field, the acoustical interfaces can be divided into the following main categories.

- Pressure acoustics—The dependent variable is the acoustic pressure p.
- Acoustic-solid interaction—The dependent variables are the pressure p and the displacement field $\mathbf{u}$ in the solid. This type of problem requires the addition of the Acoustics Module.
- Poroelastic waves—The dependent variables are the pressure p inside the saturating fluid and the total displacement $\mathbf{u}$ of the porous matrix. This type of problem requires the addition of the Acoustics Module.
- Aeroacoustics—The dependent variables are the acoustic perturbations to the background mean flow fields. In the linearized potential flow interface it is the potential ϕ for the acoustic particle-velocity field $\mathbf{v} = \nabla\phi$. In the linearized Euler interface the dependent variables are the acoustic variations in pressure p, density ρ, and velocity field $\mathbf{u}$. In the linearized Navier-Stokes they are the pressure p, velocity field $\mathbf{u}$, and temperature T. In the typical situation, the background fluid is in motion with, for example, a total velocity $\mathbf{u}_{\mathrm{tot}} = \mathbf{u}_0 + \mathbf{u}$, split into a stationary background-flow velocity $\mathbf{u}_0$ and the particle velocity $\mathbf{u}$ associated with the acoustic waves. This type of problem requires the addition of the Acoustics Module.
- Thermoacoustics—The dependent variables are the acoustic pressure p, the particle-velocity field $\mathbf{v}$, and the acoustic temperature variation T. This is a detailed acoustic model solving the full set of linearized equations for a compressible flow: Navier-Stokes (momentum conservation), continuity (mass conservation), and energy conservation equations. This type of problem requires the addition of the Acoustics Module.

These standard problems or scenarios occur frequently when analyzing acoustics:

THE RADIATION PROBLEM

A vibrating structure (a speaker, for example) radiates sound into the surrounding space. A radiation boundary condition or a PML (perfectly matched layer) is necessary to model the unbounded open domain.

THE SCATTERING PROBLEM

An incident wave impinges on a body and creates a scattered wave. A radiation boundary condition or a PML is necessary. This could be a sonar application in underwater acoustics or an analysis of the scattered sound field around a human head.

THE SOUND FIELD IN AN INTERIOR SPACE

The acoustic waves stay in a finite volume so no radiation condition is necessary. For example, this case represents the sound inside a room or a car interior. A more advanced example is the sound inside a transducer like a microphone; in this case, the acoustic field should be solved with the Thermoacoustics interface. Analysis using thermoacoustics requires the Acoustics Module.

COUPLED FLUID-ELASTIC STRUCTURE INTERACTION (STRUCTURAL ACOUSTICS)

If the radiating or scattering structure consists of an elastic material, the interaction must be considered between the body and the surrounding fluid. In the multiphysics coupling, the acoustic analysis provides a load (the sound pressure) to the structural analysis, and the structural analysis provides accelerations to the acoustic analysis.

THE TRANSMISSION PROBLEM

An incident sound wave propagates into a body, which can have different acoustic properties. Pressure and acceleration are continuous on the boundary. A typical transmission problem is that of modeling the behavior of mufflers.

AEROACOUSTICS PROBLEMS

The sound (noise) field is influenced by a background flow. This could be the propagating sound from a jet engine or the acoustic damping properties of a muffler with flow. Analysis of these types of problems require the addition of the Acoustics Module.

POROELASTIC WAVES PROBLEM

If the acoustic waves are propagating inside the saturating fluid of porous material the detailed coupling between the fluid pressure and the solid displacement need to be taken into account. In cases where only the fluid pressure is of interest, the porous material can be modeled using an equivalent fluid model. Analysis of this type of problem requires the addition of the Acoustics Module.

TRANSDUCER PROBLEMS

Transducers are devices for transformation of one form of energy to another (electrical, mechanical, or acoustical). This type of problem is common in acoustics and is a true multiphysics problem involving electric, structural, and acoustic interfaces. Typical problems of this type involve modeling loudspeakers, microphones, and piezo transducers. Analysis of these types of problems require the addition of the Acoustics Module.

Mathematical Models for Acoustic Analysis

Standard acoustic problems involve solving for the small acoustic pressure variations p on top of the stationary background pressure p_0. Mathematically this represents a linearization (small parameter expansion) around the stationary quiescent values.

The governing equations for a compressible lossless (no thermal conduction and no viscosity) fluid flow problem are the momentum equation (Euler's equation) and the continuity equation. These are given by:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p$$

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0$$

where ρ is the total density, p is the total pressure, and $\mathbf{u}$ is the velocity field. In classical pressure acoustics all thermodynamic processes are assumed reversible and adiabatic, known as an isentropic process. The small parameter expansion is performed on a stationary fluid of density ρ_0 (SI unit: $\mathrm{kg/m^3}$) and at pressure p_0 (SI unit: Pa) such that:

$$\begin{aligned} p &= p_0 + p' \\ \rho &= \rho_0 + \rho' \\ \mathbf{u} &= \mathbf{0} + \mathbf{u}' \end{aligned} \qquad \text{with} \qquad \begin{aligned} p' &\ll p_0 \\ \rho' &\ll \rho_0 \end{aligned}$$

where the primed variables represent the small acoustic variations. Inserting these into the governing equations and only retaining terms linear in the primed variables yields:

$$\frac{\partial \mathbf{u}'}{\partial t} = -\frac{1}{\rho_0}\nabla p'$$

$$\frac{\partial \rho'}{\partial t} + \rho_0(\nabla\cdot\mathbf{u}') = 0$$

One of the dependent variables, the density, is removed by expressing it in terms of the pressure using a Taylor expansion (linearization):

$$\rho' = \left.\frac{\partial \rho_0}{\partial p}\right|_s p' = \frac{1}{c_s^2}p'$$

where c_s is recognized as the (isentropic) speed of sound (SI unit: m/s) at constant entropy s. The subscripts s and 0 are dropped in the following. Finally, rearranging the equations (divergence of momentum equation inserted into the continuity equation) and dropping the primes yields the wave equation for sound waves in a lossless medium:

$$\frac{1}{\rho c^2}\frac{\partial^2 p}{\partial t^2} + \nabla\cdot\left(-\frac{1}{\rho}(\nabla p - \mathbf{q}_d)\right) = Q_m \tag{11-1}$$

The speed of sound is related to the compressibility of the fluid where the waves are propagating. The combination ρc^2 is called the *bulk modulus*, commonly denoted K (SI unit: $\mathrm{N/m^2}$). The equation is further extended with two optional source terms: the *dipole source* $\mathbf{q}_\mathrm{d}$ (SI unit: $\mathrm{N/m^3}$) and the *monopole source* Q_m (SI unit: $1/\mathrm{s^2}$).

A special case is a time-harmonic wave, for which the pressure varies with time as

$$p(\mathbf{x}, t) = p(\mathbf{x})e^{i\omega t}$$

where $\omega = 2\pi f$ (SI unit: rad/s) is the angular frequency and f (SI unit: Hz) is denoting the frequency. Assuming the same harmonic time-dependence for the source terms, the wave equation for acoustic waves reduces to an inhomogeneous Helmholtz equation:

$$\nabla\cdot\left(-\frac{1}{\rho}(\nabla p - \mathbf{q}_d)\right) - \frac{\omega^2 p}{\rho c^2} = Q_m \tag{11-2}$$

With the two source terms removed, this equation can also be treated as an eigenvalue PDE to solve for eigenmodes and eigenfrequencies.

Typical boundary conditions for the wave equation and the Helmholtz equation are sound-hard boundaries (walls), sound-soft boundaries, impedance boundary conditions, and radiation boundary conditions.

In lossy media, an additional term of first order in the time derivative needs to be introduced to model attenuation of the sound waves:

$$\frac{1}{\rho c^2}\frac{\partial^2 p}{\partial t^2} - d_{\mathrm{a}}\frac{\partial p}{\partial t} + \nabla \cdot \left(-\frac{1}{\rho}(\nabla p - \mathbf{q}_d)\right) = Q_m$$

where d_{a} is the damping coefficient. Note also that even when the sound waves propagate in a lossless medium, attenuation frequently occurs by interaction with the surroundings at the boundaries of the system.

Theory for Pressure Acoustics, Frequency Domain

The Pressure Acoustics, Frequency Domain Interface is designed for the analysis of various types of pressure acoustics problems in the frequency domain, all concerning pressure waves in a fluid. An acoustics model can be part of a larger multiphysics model that describes, for example, the interactions between structures and acoustic waves. This physics interface is suitable for modeling acoustics phenomena that do not involve fluid flow.

This physics interface solves for the acoustic pressure, p. It is available in all space dimensions—for 3D, 2D, and 1D Cartesian geometries as well as for 2D and 1D axisymmetric geometries.

These studies are discussed briefly in this section:

- Frequency Domain Study
- Eigenfrequency Study
- References for the Pressure Acoustics, Frequency Domain Interface

Frequency Domain Study

The frequency domain—or time-harmonic—formulation uses the following inhomogeneous Helmholtz equation:

$$\nabla\cdot\left(-\frac{1}{\rho_c}(\nabla p-\mathbf{q}_d)\right)-\frac{\omega^2 p}{\rho_c c_c^2}=Q_m \tag{11-3}$$

In this equation, $p = p\ (\mathbf{x},\ \omega)$ (the dependence on ω is henceforth not explicitly indicated). With this formulation compute the frequency response with a parametric sweep over a frequency range using a harmonic load.

When there is damping, ρ_c and c_c are complex-valued quantities. The available damping models and how to apply them is described in the section Damping Models

Equation 11-3 is the equation that the software solves for 3D geometries. In lower-dimensional and axisymmetric cases, restrictions on the coordinate dependence mean that the equations differ from case to case. Here is a brief summary of the situation.

2D

In 2D, the pressure is of the form

$$p(\mathbf{r})=p(x,y)e^{-ik_z z}$$

which inserted in Equation 11-3 gives

$$\nabla\cdot\left(-\frac{1}{\rho_c}(\nabla p-\mathbf{q}_d)\right)-\frac{1}{\rho_c}\left(\frac{\omega^2}{c_c^2}-k_z^2\right)p=Q_m \tag{11-4}$$

The *out-of-plane wave number*, k_z, can be set on the Pressure Acoustics page. By default its value is 0. In the mode analysis type, $-ik_z$ is used as the eigenvalue.

2D AXISYMMETRY

For 2D axisymmetric geometries the independent variables are the radial coordinate, r, and the axial coordinate, z. The only dependence allowed on the azimuthal coordinate, φ , is through a phase factor,

$$p(r,\phi,z)=p(r,z)e^{-im\varphi} \tag{11-5}$$

where m denotes the *circumferential wave number*. Because the azimuthal coordinate is periodic, m must be an integer. Just like k_z in the 2D case, m can be set on the Pressure Acoustics settings window.

As a result of Equation 11-5, the equation to solve for the acoustic pressure in 2D axisymmetric geometries becomes

$$\frac{\partial}{\partial r}\left[-\frac{r}{\rho_c}\left(\frac{\partial p}{\partial r}-q_r\right)\right]+r\frac{\partial}{\partial z}\left[-\frac{1}{\rho_c}\left(\frac{\partial p}{\partial z}-q_z\right)\right]-\left[\left(\frac{\omega}{c_c}\right)^2-\left(\frac{m}{r}\right)^2\right]\frac{rp}{\rho_c}=rQ_m$$

1D AXISYMMETRY

In 1D axisymmetric geometries,

$$p(r,\phi,z)=p(r)e^{-i(k_z z+m\varphi)}$$

leading to the radial equation

$$\frac{\partial}{\partial r}\left[-\frac{r}{\rho_c}\left(\frac{\partial p}{\partial r}-q_r\right)\right]-\left[\left(\frac{\omega}{c_c}\right)^2-\left(\frac{m}{r}\right)^2-k_z^2\right]\frac{rp}{\rho_c}=rQ_m$$

where both the circumferential wave number m, and the *axial wave number* k_z, appear as parameters.

1D

The equation for the 1D case is obtained by taking the pressure to depend on a single Cartesian coordinate, x:

$$\frac{d}{dx}\left(-\frac{1}{\rho_c}\left(\frac{dp}{dx}-q_{\mathrm{d}}\right)\right)-\frac{\omega^2}{\rho_c c_c^2}p=Q_m$$

Eigenfrequency Study

In the eigenfrequency formulation the source terms are absent, the eigenmodes and eigenfrequencies are solved for:

$$\nabla\cdot\left(-\frac{1}{\rho_c}\nabla p\right)+\frac{\lambda^2 p}{\rho_c c^2}=0 \qquad (11\text{-}6)$$

The eigenvalue λ introduced in this equation is related to the eigenfrequency f, and the angular frequency ω, through $\lambda = i2\pi f = i\omega$. Because they are independent of the pressure, the solver ignores any dipole and monopole sources unless a coupled eigenvalue problem is being solved.

Equation 11-6 applies to the 3D case. The equations solved in eigenfrequency studies in lower dimensions and for axisymmetric geometries are obtained from their time-harmonic counterparts, given in the previous subsection, by the substitution $\omega^2 \rightarrow -\lambda^2$.

Switch between specifying the eigenvalues, the eigenfrequencies, or the angular frequencies by selecting from the Eigenvalue transformation list in the solver sequence's Eigenvalue feature node's settings window.

References for the Pressure Acoustics, Frequency Domain Interface

1. D. Givoli and B. Neta, "High-order Non-reflecting Boundary Scheme for Time-dependent Waves," *J. Comput. Phys.*, vol. 186, pp. 24–46, 2004.

2. A. Bayliss, M. Gunzburger, and E. Turkel, "Boundary Conditions for the Numerical Solution of Elliptic Equations in Exterior Regions," *SIAM J. Appl. Math.*, vol. 42, no. 2, pp. 430–451, 1982.

3. A.B. Bauer, “Impedance Theory and Measurements on Porous Acoustic Liners,” *J. Aircr.*, vol. 14, pp. 720–728, 1977.

4. S. Temkin, *Elements of Acoustics*, Acoustical Society of America, 2001.

The Pressure Acoustics, Frequency Domain Interface

The **Pressure Acoustics, Frequency Domain (acpr)** interface (), found under the **Pressure Acoustics** branch () when adding a physics interface, is used to compute the pressure variation for propagation of acoustic waves in fluids at quiescent background conditions. It is suited for all frequency-domain simulations with harmonic variations of the pressure field.

The physics interface can be used for linear acoustics described by a scalar pressure variable. It includes domain conditions to model losses in a homogenized way, so-called fluid models for porous materials, as well as losses in narrow regions. Domain features also include background incident acoustic fields, as well as domain monopole and dipole sources. The plane wave attenuation behavior of the acoustic waves may be entered as a user-defined quantity, or defined to be bulk viscous and thermal losses.

The physics interface solves the Helmholtz equation in the frequency domain for given frequencies, or as an eigenfrequency or modal analysis study.

An acoustics model can be part of a larger multiphysics model that describes, for example, the interactions between structures and acoustic waves. This physics interface is suitable for modeling acoustics phenomena that do not involve fluid flow.

The sound pressure p, which is solved for in pressure acoustics, represents the acoustic variations (or excess pressure) to the ambient pressure. In the absence of flow, the ambient pressure is simply the static absolute pressure.

When the geometrical dimensions of the acoustic problems are reduced from 3D to 2D (planar symmetry or axisymmetric) or to 1D axisymmetric, it is possible to specify an out-of-plane wave number k_z and a circumferential wave number m, when applicable. The wave number used in the equations k_{eq} contains both the ordinary wave number k as well as the out-of-plane wave number and circumferential wave number, when applicable.

The following table lists the names and SI units for the most important physical quantities in the Pressure Acoustics, Frequency Domain interface:

TABLE 11-1: PRESSURE ACOUSTICS, FREQUENCY DOMAIN INTERFACE PHYSICAL QUANTITIES

QUANTITY	SYMBOL	SI UNIT	ABBREVIATION
Pressure	p	pascal	Pa
Density	ρ	kilogram/meter^3	kg/m^3
Frequency	f	hertz	Hz
Wave number	k	1/meter	1/m
Dipole source	$\mathbf{q}_\mathrm{d}$	newton/meter^3	N/m^3
Monopole source	Q_m	1/second^2	1/s^2
Speed of sound	c	meter/second	m/s
Acoustic impedance	Z	pascal-second/meter	Pa·s/m
Normal acceleration	a_n	meter/second^2	m/s^2
Source location	r_0	meter	m
Wave direction	$\mathbf{n}_\mathrm{k}$	(dimensionless)	1

In the following descriptions of the functionality in this physics interface, the subscript c in ρ_c and c_c (the density and speed of sound, respectively) denotes that these can be complex-valued quantities in models with damping.

When this physics interface is added, these default nodes are also added to the **Model Builder**— **Pressure Acoustics Model**, **Sound Hard Boundary (Wall)**, and **Initial Values**.

Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and point conditions. You can also right-click **Pressure Acoustics** to select physics from the context menu.

Physics Nodes—Equation Section

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `acpr`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define a sound pressure field and the associated acoustics equation. To choose specific domains, select **Manual** from the **Selection** list.

EQUATION

Expand the **Equation** section to see the equations solved for with the **Equation form** specified. The default selection is **Equation form** is set to **Study controlled**. The available studies are selected under **Show equations assuming**.

When the **Equation form** is set to **Study controlled**, the scaling and non-reflecting boundary settings are optimized for the numerical performance of the different solvers.

To display the **Pressure Acoustics Equation Settings** section for 2D and 1D models, click to expand the **Equation** section, then select **Frequency domain** as the **Equation form** and enter the settings as described in Scaling Factor and Non-reflecting Boundary Condition Approximation.

PRESSURE ACOUSTICS EQUATION SETTINGS

For 1D axisymmetric models, the **Circumferential wave number** m (dimensionless) default is 0 and the **Out-of-plane wave number** k_z (SI unit: rad/m) default is 0 rad/m. Enter different values or expressions as required.

For 2D axisymmetric models, the **Circumferential wave number** m (dimensionless) default is 0. Enter a different value or expression as required.

For 2D models, the **Out-of-plane wave number** k_z (SI unit: rad/m) default is 0 rad/m. Enter a different value or expression as required.

Scaling Factor and Non-reflecting Boundary Condition Approximation

For all component dimensions, and if required, click to expand the **Equation** section, then select **Frequency domain** as the **Equation form** and enter the settings as described below.

The default **Scaling factor** Δ is $1/\omega^2$ and **Non-reflecting boundary condition approximation** is **Second order**. These values correspond to the equations for a Frequency Domain study when the equations are study controlled.

To get the equations corresponding to an Eigenfrequency study, change the **Scaling factor** $\varDelta$ to 1 and the **Non-reflecting boundary conditions approximation** to **First order**.

SOUND PRESSURE LEVEL SETTINGS

The zero level on the dB scale varies with the type of fluid. That value is a reference pressure that corresponds to 0 dB. This variable occurs in calculations of the sound pressure level L_p based on the root mean square (rms) pressure p_rms, such that

$$L_\mathrm{p} = 20\log\left(\frac{p_\mathrm{rms}}{p_\mathrm{ref}}\right) \qquad \text{with} \qquad p_\mathrm{rms} = \sqrt{\frac{1}{2}pp^*}$$

where p_ref is the reference pressure and the star (*) represents the complex conjugate. This is an expression valid for the case of harmonically time-varying acoustic pressure p.

Based on the fluid type, select a **Reference pressure for the sound pressure level**. Select:

- **Use reference pressure for air** to use a reference pressure of 20 μPa ($20\cdot10^{-6}$ Pa).
- **Use reference pressure for water** to use a reference pressure of 1 μPa ($1\cdot10^{-6}$ Pa).
- **User-defined reference pressure** to enter a reference pressure $p_\mathrm{ref,\,SPL}$ (SI unit: Pa). The default value is the same as for air, 20 μPa.

DEPENDENT VARIABLES

This physics interface defines one dependent variable (field), the **Pressure** p. If required, edit the name, which changes both the field name and the dependent variable name. If the new field name coincides with the name of another pressure field in the model, the interfaces share degrees of freedom and dependent variable name. The new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select **Quadratic** (the default), **Linear**, **Cubic**, **Quartic**, or **Quintic** for the **Pressure**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for the Pressure Acoustics, Frequency Domain Interface
- Theory for Pressure Acoustics, Frequency Domain

Eigenmodes of a Room: model library path **COMSOL_Multiphysics/Acoustics/eigenmodes_of_room**

Domain, Boundary, Edge, Point, and Pair Nodes for the Pressure Acoustics, Frequency Domain Interface

The Pressure Acoustics, Frequency Domain Interface has these domain, boundary, edge, point, and pair nodes, listed in alphabetical order, available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

- Continuity
- Cylindrical Wave Radiation
- Dipole Source
- Destination Selection
- Impedance
- Incident Pressure Field
- Interior Sound Hard Boundary (Wall)
- Initial Values
- Monopole Source
- Normal Acceleration
- Periodic Condition
- Plane Wave Radiation
- Pressure Acoustics
- Pressure
- Sound Hard Boundary (Wall)
- Sound Soft Boundary
- Spherical Wave Radiation
- Symmetry

Continuity in the total pressure is the default condition on interior boundaries.

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Pressure Acoustics

The **Pressure Acoustics** node adds the equations for time-harmonic and eigenfrequency acoustics modeling in the frequency domain. In the settings window, define the properties for the acoustics model and model inputs including temperature.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to compute the acoustic pressure field and the equation that defines it, or select **All domains** as required.

PRESSURE ACOUSTICS MODEL

The default **Fluid Model** for pressure acoustics is a **Linear elastic** fluid. By default the values for the **Density** ρ (SI unit: $\mathrm{kg/m^3}$) and the **Speed of sound** c (SI unit: m/s) are taken **From material**. Select **User defined** to enter other values for these properties.

Monopole Source

Use the **Monopole Source** node to add a the domain source term Q_m to the governing equation. A monopole source added to a domain has a uniform strength in all directions. In advanced models this source term can, for example, be used to represent a domain heat source causing pressure variations. Add this node from the **More>** submenu.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

MONOPOLE SOURCE

Enter a **Monopole source** Q_m (SI unit: $\mathrm{1/s^2}$). The default is $0\ \mathrm{1/s^2}$.

Dipole Source

Use the **Dipole Source** node to add the domain source term $\mathbf{q}_\mathrm{d}$ to the governing equation. This source is typically stronger in two opposite directions. In advanced models this term can, for example, be used to represent a uniform constant background flow convecting the sound field. Add this node from the **More>** submenu.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

DIPOLE SOURCE

Enter coordinates for the **Dipole source** $\mathbf{q}_\mathrm{d}$ (SI unit: $\mathrm{N/m^3}$). These are the individual components of the dipole source vector. The defaults are $0\ \mathrm{N/m^3}$.

Initial Values

The **Initial Values** node adds initial values for the sound pressure and the pressure time derivative that can serve as an initial guess for a nonlinear solver. If more than one initial value is needed, from the **Physics** toolbar click to add more **Initial Values** nodes.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the initial values for the **Pressure** p (SI unit: Pa) and the **Pressure, first time derivative**, $\partial p/\partial t$ (SI unit: Pa/s). The defaults are 0 Pa and 0 Pa/s, respectively.

Sound Hard Boundary (Wall)

The **Sound Hard Boundary (Wall)** adds a boundary condition for a sound hard boundary or wall, which is a boundary at which the normal component of the acceleration is zero:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q}_d)\right) = 0$$

For zero dipole source and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary:

$$\frac{\partial p}{\partial \mathbf{n}} = 0$$

Sound-hard boundaries are available for all study types. Note that this condition is identical to the Symmetry condition.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

Normal Acceleration

The **Normal Acceleration** adds an inward normal acceleration a_n:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q}_d)\right) = a_\mathrm{n}$$

Alternatively, specify the acceleration $\mathbf{a}_0$ of the boundary. The part in the normal direction is used to define the boundary condition:

$$\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q}_d)\right) = \mathbf{n}\cdot\mathbf{a}_0$$

This feature represents an external source term. It can also be used to manually couple acoustics with a structural analysis for modeling acoustic-structure interaction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

NORMAL ACCELERATION

Select a **Type**—**Inward Acceleration** (the default) or **Acceleration**.

- If **Inward Acceleration** is chosen, enter the value of the **Inward acceleration** a_n (SI unit: m/s^2). The default is 0 m/s^2. Use a positive value for inward acceleration or a negative value for outward acceleration.
- If **Acceleration** is chosen, enter values for the components of the **Acceleration** $\mathbf{a}_0$ (SI unit: m/s^2). The defaults are 0 m/s^2.

Sound Soft Boundary

The **Sound Soft Boundary** adds a boundary condition for a sound soft boundary, where the acoustic pressure vanishes: $p = 0$. It is an appropriate approximation for a liquid-gas interface and in some cases for external waveguide ports.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. If the node is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Pressure

The **Pressure** node creates a boundary condition that acts as a pressure source at the boundary, which means a constant acoustic pressure p_0 is specified and maintained at the boundary: $p = p_0$. In the frequency domain, p_0 is the amplitude of a harmonic pressure source.

The node is also available from the **Pairs** submenu as an option at interfaces between parts in an assembly.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. If the node is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PRESSURE

Enter the value of the **Pressure** p_0 (SI unit: Pa). The default is 0 Pa.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

These are the same settings as for Sound Soft Boundary.

Impedance

The **Impedance** node adds an impedance boundary condition, which is a generalization of the sound-hard and sound-soft boundary conditions:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_c}(\nabla p_{\mathrm{t}}-\mathbf{q}_{\mathrm{d}})\right) = -\frac{i\omega p_{\mathrm{t}}}{Z_i}$$

Here Z_{i} is the acoustic input impedance of the external domain and it has the unit of a specific acoustic impedance. From a physical point of view, the acoustic input impedance is the ratio between the local pressure and local normal particle velocity.

The **Impedance** boundary condition is a good approximation for a locally reacting surface—a surface for which the normal velocity at any point depends only on the pressure at that exact point.

In the two opposite limits $Z_i \to \infty$ and $Z_i \to 0$, this boundary condition is identical to the Sound Hard Boundary (Wall) condition and the Sound Soft Boundary condition, respectively.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

IMPEDANCE

Enter the value of the **Impedance** Z_{i} (SI unit: Pa·s/m). The default value is set to the specific impedance of air 1.2 kg/m^3·343 m/s.

Symmetry

The **Symmetry** node adds a boundary condition where there is symmetry in the pressure. Use this condition to reduce the size of a model by cutting it in half where there are symmetries. In pressure acoustics this boundary condition is mathematically identical to the Sound Hard Boundary (Wall) condition.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. If the node is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Plane Wave Radiation

The **Plane Wave Radiation** node adds a radiation boundary condition for a plane wave. If required, from the **Physics** toolbar, add an Incident Pressure Field to model an incoming wave. This radiation condition allows an outgoing plane wave to leave the modeling domain with minimal reflections, when the angle of incidence is near to normal.

The *plane wave* type is suitable for both far-field boundaries and ports. Because many waveguide structures are only interesting in the plane-wave region, it is particularly relevant for ports. When using the radiation condition on an open far-field boundary it is recommended to construct the boundary such that the incidence angle is near to normal, this of course requires a priory knowledge of the problem and the solution.

An estimate of the reflection coefficient R_s, for the spurious waves reflecting off the plane wave radiation boundary, is, for incident plane waves at angle θ, given by the expression:

$$R_s = \left|\frac{\cos\theta - 1}{\cos\theta + 1}\right|^N$$

where N is the order of the boundary condition (here 1 or 2). So at normal incidence ($\theta = 0$) there are no spurious reflections, while, for example, at an incidence angle of 30^0 for $N = 2$ (plane wave radiation in the frequency domain) the amplitude of the spurious reflected wave is 0.5 % of the incident.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Acoustics of a Muffler: model library path **COMSOL_Multiphysics/Acoustics/automotive_muffler**

Spherical Wave Radiation

The **Spherical Wave Radiation** node adds a radiation boundary condition for a spherical wave, for which you define the source location. If required, from the **Physics** toolbar add an Incident Pressure Field to model an incoming wave. This radiation condition allows an outgoing spherical wave to leave the modeling domain with minimal reflections. The geometry of the modeling domain should be adapted to have the outgoing spherical waves coincide with the boundary, this is in order to minimize spurious reflections.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

SPHERICAL WAVE RADIATION

Enter coordinates for the **Source location** r_0 (SI unit: m). The defaults are 0 m.

Cylindrical Wave Radiation

The **Cylindrical Wave Radiation** node adds a radiation boundary condition for a cylindrical wave, for which you define the source location and the source axis direction If required, from the **Physics** toolbar add an Incident Pressure Field to model an incoming wave. This radiation condition allow an outgoing cylindrical wave to leave the modeling domain with minimal reflections. The geometry of the modeling domain should be adapted to have the outgoing cylindrical waves coincide with the boundary, this is in order to minimize spurious reflections.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

CYLINDRICAL WAVE RADIATION

Enter coordinates for the **Source location** r_0 (SI unit: m) (the defaults are 0 m) and the **Source axis** direction $r_{\rm axis}$ (dimensionless) (the defaults are 0).

Incident Pressure Field

The **Incident Pressure Field** node is a subnode to all non-reflecting boundary conditions (plane, cylindrical, or spherical wave radiation). From the **Physics** toolbar, add Plane Wave Radiation, Spherical Wave Radiation, or Cylindrical Wave Radiation nodes. If the incident pressure field $p_{\rm i}$ is a predefined plane wave, it is of the type:

$$p_{\rm i} = p_0 e^{-i(\mathbf{k}\cdot\mathbf{r})} = p_0 e^{-ik_{\rm eq}\left(\frac{\mathbf{r}\cdot\mathbf{e}_k}{\|\mathbf{e}_k\|}\right)}$$

where p_0 is the wave amplitude, $\mathbf{k}$ is the wave vector (with amplitude $k_{\rm eq}=|\mathbf{k}|$ and wave direction vector $\mathbf{e}_k$), and $\mathbf{r}$ is the location on the boundary. The incident pressure field can also be a user-defined value or expression.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to include an incident pressure field $p_{\rm i}$ in the boundary condition. By default, this feature node inherits the selection from its parent node, and only a selection that is a subset of the parent node's selection can be used.

INCIDENT PRESSURE FIELD

From the **Incident pressure field type** list, select **Plane wave** to define an incident pressure field of plane wave type. Then enter a **Pressure amplitude** p_0 (SI unit: Pa) (the default is 0 Pa) and **Wave direction** $\mathbf{e}_k$ (SI unit: m).

Select **User defined** to enter the expression for the **Incident pressure field** $p_{\rm i}$ (SI unit: Pa) as a function of space. The default is 0 Pa.

Periodic Condition

The **Periodic Condition** node adds a periodic boundary condition that can be used to reduce the model size by using symmetries and periodicities in the geometry and physics being modeled.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. The software automatically identifies the boundaries as either source boundaries or destination boundaries.

This feature works well for cases like opposing parallel boundaries. In other cases use a **Destination Selection** subnode to control the destination. By default it contains the selection that COMSOL Multiphysics identifies.

PERIODICITY SETTINGS

Select a **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

CONSTRAINT SETTINGS

These are the same settings as for Sound Soft Boundary.

- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

Interior Sound Hard Boundary (Wall)

The **Interior Sound Hard Boundary (Wall)** node adds a boundary condition for a sound hard boundary or wall on interior boundaries. A sound-hard boundary is a boundary at which the normal component of the acceleration is zero:

$$-\mathbf{n}\cdot -\left(\frac{1}{\rho_c}(\nabla p_\mathrm{t}-\mathbf{q}_\mathrm{d})\right)_1 = 0 \qquad -\mathbf{n}\cdot\left(\left(-\frac{1}{\rho_c}\right)(\nabla p_\mathrm{t}-\mathbf{q}_\mathrm{d})\right)_2 = 0$$

where the subscripts 1 and 2 represent the two sides of the boundary. For zero dipole charge and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary. On an interior sound hard boundary the pressure is not continuous but is treated as a so-called slit feature.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Axial Symmetry

The **Axial Symmetry** node is a default node added for all 2D and 1D axisymmetric models. The boundary condition is active on all boundaries on the symmetry axis.

BOUNDARY SELECTION

The boundaries section shows on which boundaries the node is active. All boundaries on the symmetry axis are automatically selected.

Continuity

Continuity is available as an option at interfaces between parts in a pair.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

This condition gives continuity in total pressure and in the normal acceleration over the pair (subscripts 1 and 2 in the equation refer to the two sides in the pair):

$$-\mathbf{n}\cdot\left[-\left(\frac{1}{\rho_c}(\nabla p_\mathrm{t}-\mathbf{q}_\mathrm{d})\right)_1 - \left(-\frac{1}{\rho_c}(\nabla p_\mathrm{t}-\mathbf{q}_\mathrm{d})\right)_2\right] = 0$$

BOUNDARY SELECTION

This list cannot be edited. It shows the boundaries in the selected pairs.

PAIR SELECTION

When this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

These are the same settings as for Sound Soft Boundary.

12

Chemical Species Transport

This chapter explains how to use the Transport of Diluted Species interface, found under the **Chemical Species Transport** branch () when adding a physics interface, to model and simulate mass transfer by diffusion and convection based on Fick's law of diffusion.

Theory for Transport of Diluted Species

The Transport of Diluted Species Interface provides a predefined modeling environment for studying the evolution of chemical species transported by diffusion and convection. The physics interface assumes that all species present are dilute; that is, that their concentration is small compared to a solvent fluid or solid. As a rule of thumb, a mixture containing several species can be considered dilute when the concentration of the solvent is more than 90 mol%. Due to the dilution, mixture properties such as density and viscosity can be assumed to correspond to those of the solvent.

Fick's law governs the diffusion of the solutes, dilute mixtures or solutions. The **Transport of Diluted Species** interface supports the simulations of chemical species transport by convection and diffusion in 1D, 2D, and 3D as well as for axisymmetric models in 1D and 2D.

Mass Balance Equation

The default node attributed to the **Transport of Diluted Species** interface models chemical species transport through diffusion and convection and implements the mass balance equation:

$$\frac{\partial c}{\partial t}+\mathbf{u}\cdot\nabla c = \nabla\cdot(D\nabla c)+R \tag{12-1}$$

Equation 12-1 includes the following quantities:

- c is the concentration of the species (SI unit: mol/m^3)
- D denotes the diffusion coefficient (SI unit: m^2/s)
- R is a reaction rate expression for the species (SI unit: mol/(m^3·s))
- $\mathbf{u}$ is the velocity vector (SI unit: m/s)

The first term on the left-hand side of Equation 12-1 corresponds to the accumulation (or indeed consumption) of the species.

The second term accounts for the convective transport due to a velocity field $\mathbf{u}$. This field can be expressed analytically or be obtained from coupling this physics to one that describes fluid flow (momentum balance). To include convection in the mass balance equation, an expression that includes the space and time variables, or the velocity vector component variable names from a fluid-flow interface, can be entered into the appropriate field. The velocity fields from existing fluid-flow interfaces are available directly as predefined fields (model inputs) for multiphysics couplings.

On the right-hand side of the mass balance equation (Equation 12-1), the first term describes the diffusive transport, accounting for the interaction between the dilute species and the solvent. A field for the diffusion coefficient is available, and any expression containing other variables such as pressure and temperature can be entered here. The node has a matrix that you can use to describe anisotropic diffusion coefficients.

Finally, the second term on the right-hand side of Equation 12-1 represents a source or sink term, typically due to a chemical reaction. In order for the chemical reaction to be specified, another node must be added to the **Transport of Diluted Species** interface—the **Reaction** node, which has a field for specifying a reaction equation using the variable names of all participating species.

Convective Term Formulation

The default node attributed to The Transport of Diluted Species Interface assumes chemical species transport through diffusion and convection and implements the mass balance equation in Equation 12-1.

There are two ways to present a mass balance where chemical species transport occurs through diffusion and convection. These are the non-conservative and conservative formulations of the convective term:

$$\text{non-conservative: } \frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot (D \nabla c) + R \qquad (12\text{-}2)$$

$$\text{conservative: } \frac{\partial c}{\partial t} + \nabla \cdot (c\mathbf{u}) = \nabla \cdot (D \nabla c) + R \qquad (12\text{-}3)$$

and each is treated slightly differently by the solver algorithms. In these equations D (SI unit: $\mathrm{m^2/s}$) is the diffusion coefficient, R (SI unit: $\mathrm{mol/(m^3 \cdot s)}$) is a production or consumption rate expression, and $\mathbf{u}$ (SI unit: m/s) is the solvent velocity field. The diffusion process can be anisotropic, in which case D is a tensor.

If the conservative formulation is expanded using the chain rule, then one of the terms from the convection part, $c\nabla \cdot \mathbf{u}$, would equal zero for an incompressible fluid and would result in the non-conservative formulation above. This is in fact the default formulation in this physics interface and ensures that nonphysical source terms do not emerge from a solution for the flow field.

To switch between the two formulations, click the **Show** button ( ) and select **Advanced Physics Options**.

Solving a Diffusion Equation Only

Remove the convection term from Equation 12-2 and Equation 12-3 by clearing the Convection check box in the Transport Mechanisms section for The Transport of Diluted Species Interface. The equation then becomes

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c) + R$$

Mass Sources for Species Transport

There are two types of mass sources in the Transport of Diluted Species interface: point sources and line sources.

These features require at least one of the following licenses: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

POINT SOURCE

A point source is theoretically formed by assuming a mass injection/ejection, $\dot{Q}_\mathrm{c}$ (SI unit: $\mathrm{mol/(m^3 \cdot s)}$), in a small volume δV and then letting the size of the volume tend to zero while keeping the total mass flux constant. Given a point source strength, $\dot{q}_{\mathrm{p,c}}$ (SI unit: mol/s), this can be expressed as

$$\lim_{\delta V \to 0} \int_{\delta V} \dot{Q}_\mathrm{c} = \dot{q}_{\mathrm{p,c}} \qquad (12\text{-}4)$$

An alternative way to form a point source is to assume that mass is injected/extracted through the surface of a small object. Letting the object surface area tend to zero while keeping the mass flux constant, results in the same point source. For this alternative approach, effects resulting from the physical object’s volume need to be neglected.

The weak contribution

$$\dot{q}_{\mathrm{p,c}}\mathrm{test}(c)$$

is added at a point in the geometry. As can be seen from Equation 12-4, $\dot{Q}_{\mathrm{c}}$ must tend to plus or minus infinity as δV tends to zero. This means that in theory the concentration also tends to plus or minus infinity.

Observe that “point” refers to the physical representation of the source. A point source can therefore only be added to points in 3D models and to points on the symmetry axis in 2D axisymmetry models. Other geometrical points in 2D models represent physical lines.

The finite element representation of Equation 12-4 corresponds to a finite concentration at a point with the effect of the point source spread out over a region around the point. The size of the region depends on the mesh and on the strength of the source. A finer mesh gives a smaller affected region, but also a more extreme concentration value. It is important not to mesh too finely around a point source since this can result in unphysical concentration values. It can also have a negative effect on the condition number for the equation system.

LINE SOURCE

A line source can theoretically be formed by assuming a source of strength $\dot{Q}_{\mathrm{l,c}}$ (SI unit: mol/(m^3·s)), located within a tube with cross-section δS and then letting δS tend to zero while keeping the total mass flux per unit length constant. Given a line source strength, $\dot{q}_{\mathrm{l,c}}$ (SI unit: mol/(m·s)), this can be expressed as

$$\lim_{\delta S \to 0} \int_{\delta S} \dot{Q}_{\mathrm{l,c}} = \dot{q}_{\mathrm{l,c}} \tag{12-5}$$

As in the point source case, an alternative approach is to assume that mass is injected/extracted through the surface of a small object. This results in the same mass source, but requires that effects resulting from the physical object’s volume are neglected.

The weak contribution

$$\dot{q}_{\mathrm{l,c}}\mathrm{test}(c)$$

is added on lines in 3D or at points in 2D (which represent cut-through views of lines). Line sources can also be added on the axisymmetry line in 2D axisymmetry models. It cannot, however, be added on geometrical lines in 2D since those represent physical planes.

As with a point source, it is important not to mesh too finely around the line source.

For feature node information, see Line Mass Source and Point Mass Source .

For the Reacting Flow in Porous Media, Diluted Species interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, these shared physics nodes are renamed as follows:

- The **Line Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Line Source**) and one for the species (**Species Line Source**).
- The **Point Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Point Source**) and one for the species (**Species Point Source**).

Crosswind Diffusion

Transport of diluted species applications can often result in models with very high cell Péclèt number—that is, systems where convection or migration dominates over diffusion. Streamline diffusion and crosswind diffusion are of paramount importance to obtain physically reasonable results. The Transport of Diluted Species interface provides two crosswind diffusion options using different formulations. Observe that crosswind diffusion makes the equation system nonlinear even if the transport equation is linear.

DO CARMO AND GALEÃO

This is the formulation described in Numerical Stabilization. The method reduces over- and undershoots to a minimum, even for anisotropic meshes.

In some cases, the resulting nonlinear equation system can be difficult to converge. This can happen when the cell Péclèt number is very high and the model contains many thin layers such as contact discontinuities. You then have three options:

- Refine the mesh, especially in regions with thin layers.
- Use a nonlinear solver with a constant damping factor less than one.
- Switch to the Codina crosswind formulation.

CODINA

The Codina formulation is described in Ref. 1. It adds diffusion strictly in the direction orthogonal to the streamline direction. Compared to the do Carmo and Galeão formulation, the Codina formulation adds less diffusion but is not as efficient at reducing over- and undershoots. It also does not work as well for anisotropic meshes. The advantage is that the resulting nonlinear system is easier to converge and that under-resolved gradients are less smeared out.

Reference

1. R. Codina, “A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 110, pp. 325–342, 1993.

The Transport of Diluted Species Interface

Mass transfer is an important part of chemical engineering because this is the field that considers the conversion of one type of substance into another. A lot of this occurs through chemical reactions, although separation and other unit operations are an important part. You can use the Transport of Diluted Species interface to model transport of a diluted species in chemical systems by convection and diffusion.

In the Transport of Diluted Species interface, Fick's law describes the diffusive transport in the flux vector. Fick's law is adequate when the diffusing species is dilute with respect to a solvent. Assuming a binary mixture of solute A in solvent B, concentrations of up to 10 mol% of A can be considered dilute.

The optional Chemical Reaction Engineering Module has an extension of this physics interface for modeling multicomponent convection, diffusion, and migration (electrokinetic flow).

The **Transport of Diluted Species (chds)** interface (), found under the **Chemical Species Transport** branch () when adding a physics interface, is used to compute the concentration field of a dilute solute in a solvent. Transport and reactions of the species dissolved in a gas, liquid or solid can be computed. The driving forces for transport can be diffusion by Fick's law, convection, when coupled to fluid flow, and migration, when coupled to an electric field.

The physics interface supports simulation of transport by convection and diffusion in 1D, 2D, and 3D as well as for axisymmetric models in 1D and 2D. The dependent variable is the molar concentration, c.

When this physics interface is added, the following default nodes are also added in the **Model Builder**—**Convection and Diffusion**, **No Flux**, and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and rate expression terms. You can also right-click **Transport of Diluted Species** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to these variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first physics interface in the component) is `chds`.

DOMAIN SELECTION

The default setting is to have **All domains** in the component define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

There might be a domain in the model that is not described by mass transfer, such as a reactor's solid wall. In this case, remove that domain selection from here.

TRANSPORT MECHANISMS

Diffusion is always included. By default, the **Convection** check box is selected under **Additional transport mechanisms**. Click to clear the check box to only model diffusion. The Dynamic Transport Feature Node name changes according to the selection made.

If you have the Batteries & Fuel Cells Module, Electrodeposition Module, Corrosion Module, Microfluidics Module, or the Chemical Reaction Engineering Module, you can also choose to model using the **Migration in electric field** option.

DEPENDENT VARIABLES

The dependent variable is **Concentration** c. The species are dependent variables, and their names must be unique with respect to all other dependent variables in the component.

CONSISTENT AND INCONSISTENT STABILIZATION

To display these sections, click the **Show** button ( ) and select **Stabilization**. Any settings unique to this physics interface are listed below.

- When the **Crosswind diffusion** check box is selected, a weak term that reduces spurious oscillations is added to the transport equation. The resulting system is nonlinear. There are two options for the **Crosswind diffusion type**:
  - **Do Carmo and Galeão**—the default option. This type of crosswind diffusion reduces undershoots and overshoots to a minimum but can in rare cases give equation systems that are difficult to fully converge.
  - **Codina**. This options is less diffusive compared to the Do Carmo and Galeão option but can result in more undershoots and overshoots. It is also less effective for anisotropic meshes. The Codina option activates a text field for the **Lower gradient limit** g_{lim} (SI unit: $\mathrm{mol/m^4}$). It defaults to `0.1[mol/m^3)/chds.helem`, where `chds.helem` is the local element size.
- For both consistent stabilization methods, select an **Equation residual. Approximate residual** is the default setting and it means that derivatives of the diffusion tensor components are neglected. This setting is usually accurate enough and is computationally faster. If required, select **Full residual** instead.

ADVANCED SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed. Select a **Convective term**—**Non-conservative form** (the default) or **Conservative form**. The conservative formulation should be used for compressible flow.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**.

- Select an element order (shape function order) for the **Concentration**—**Linear** (the default), **Quadratic**, **Cubic**, or **Quartic**.
- The **Compute boundary fluxes** check box is selected by default so that COMSOL computes predefined accurate boundary flux variables. The computation of the following boundary flux variables changes so that:
  - `ndflux_c` (where `c` is the dependent variable for the concentration) is the normal diffusive flux and corresponds to the accurate boundary flux when diffusion is the only contribution to the flux term. When

transport in an electric field is included or when the conservative form of the equation is used, `ndflux_c` is instead computed directly from the dependent variable.

- `ntflux_c` (where `c` is the dependent variable for the concentration) is the normal total flux and corresponds to the accurate boundary flux plus additional transport terms, for example, the convective flux when you use the non-conservative form.

If you clear the **Compute boundary fluxes** check box, COMSOL instead compute the flux variables from the dependent variables using extrapolation.

- The **Apply smoothing to boundary fluxes** check box is selected by default. The smoothing can provide a more well-behaved flux value close to singularities.
- Specify the **Value type when using splitting of complex variables**—**Real** (the default) or **Complex**.

- Convective Term Formulation
- Show More Physics Options
- Domain, Boundary, and Pair Nodes for the Transport of Diluted Species Interface
- Theory for Transport of Diluted Species

- Effective Diffusivity in Porous Materials: model library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**
- Thin-Layer Diffusion: model library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**

Domain, Boundary, and Pair Nodes for the Transport of Diluted Species Interface

The Transport of Diluted Species Interface has the following domain, boundary, and pair nodes, listed in alphabetical order, available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

- Concentration
- Dynamic Transport Feature Node
- Flux
- Flux Discontinuity
- Inflow
- Initial Values
- Line Mass Source
- No Flux (the default boundary condition)
- Outflow
- Periodic Condition
- Point Mass Source
- Reactions
- Symmetry
- Thin Diffusion Barrier
- Thin Impermeable Barrier

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node that is valid on boundaries representing the symmetry axis.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

A boundary pair occurs when the solutions on two separate surfaces within a simulation are related (such as when two components in an assembly are in contact).

Dynamic Transport Feature Node

This node is dynamically dependent on the Transport Mechanisms chosen in the Transport of Diluted Species interface (convection and diffusion), and includes only the input fields required by the activated transport mechanisms. It has all the equations defining transport of diluted species as well as inputs for the material properties. The name of the node also changes the activated transport mechanisms, and can be one of the following:

- Diffusion
- Convection and Diffusion

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define material properties and other parameters that govern the transport equations, or select **All domains** as required.

If there is more than one type of domain, with different material properties, it might be necessary to deselect some of the domains. These are then defined in an additional Convection and Diffusion node.

MODEL INPUTS

If transport by convection is active the velocity field of the solvent needs to be specified as a model input.

Select the source of the **Velocity field** $\mathbf{u}$:

- Select **User defined** to enter values or expressions for the velocity components (SI unit: m/s) in the fields or table that appears below the drop-down menu. This input option is always available.
- Select the velocity field solved by a fluid flow interface that has also been added to the model. These physics interfaces have their own tags or **Interface Identifier**, and they are available to choose in the **Velocity field** drop-down menu, if they are active in the domains. This lists the variable names related to the fluid flow interface in the table underneath the drop-down menu.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used to define directions for an anisotropic diffusion coefficient.

DIFFUSION

Select an option from the **Bulk material** list. The default is **None**.

Under **Diffusion coefficient** select the appropriate scalar or tensor type to describe the diffusion transport—**Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic**—then enter the values for D_c (SI unit: m^2/s) in the corresponding field.

Initial Values

The **Initial Values** node specifies the initial values for the concentration of each species. These serve as an initial guess for a stationary solver or as initial conditions for a transient simulation.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

If there are several types of domains with different initial values defined, it might be necessary to remove some domains from the selection. These are then defined in an additional **Initial Values** node.

INITIAL VALUES

Enter a value or expression for the initial value of the **Concentration** c. The default value is 0 mol/m^3.

Line Mass Source

The **Line Mass Source** feature models mass flow originating from a tube region with infinitely small radius.

This feature requires at least one of the following licenses: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

For the Reacting Flow in Porous Media, Diluted Species interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, the **Line Mass Source** node is available in two versions, one for the fluid flow (**Fluid Line Source**) and one for the species (**Species Line Source**).

SELECTION

The **Line Mass Source** feature is available for all dimensions, but the applicable selection differs between the dimensions.

MODEL DIMENSION	APPLICABLE GEOMETRICAL ENTITY
2D	Points
2D Axisymmetry	Points not on the symmetry axis and the symmetry axis
3D	Edges

SPECIES SOURCE

Enter the source strength, $\dot{q}_{l,c}$, for each species (SI unit: mol/(m·s)). A positive value results in species injection from the line into the computational domain, and a negative value means that the species is removed from the computational domain.

Line sources located on a boundary affect the adjacent computational domains. This effect makes the physical strength of a line source located in a symmetry plane twice the given strength.

Mass Sources for Species Transport

Reactions

Use the **Reactions** node to account for the consumption or production of species. Define the rate expression as required, which display on the right-hand side of the species transport equations in the **Convection and Diffusion** node. In other words, these two nodes are integrated.

DOMAIN SELECTION

From the **Selection** list, choose the domains on which to define rate expression that govern the source term in the transport equations.

If there are several types of domains, with subsequent and different reactions occurring within them, it might be necessary to remove some domains from the selection. These are then defined in an additional **Reactions** node.

REACTIONS

Add a rate expression, R_c (SI unit: mol/(m^3·s)), for the species to be solved for. Enter a value or expression in the field .

No Flux

The **No Flux** node, which is the default boundary condition on exterior boundaries, represents boundaries where no mass flows in or out of the boundaries. Hence, the total flux is zero:

$$-\mathbf{n}\cdot(c\mathbf{u}-D\nabla c) = N_0$$

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

NO FLUX

Select **Apply for all species** (the default) to specify that the boundary is completely impervious for all species. If **Apply for...** is selected, click to select the check box for the species to specify the condition.

Concentration

The **Concentration** node adds a boundary condition for the species concentration. For example, a $c = c_0$ condition specifies the concentration of species c.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to apply the condition. An identity pair has to be created first. Ctrl-click to deselect.

CONCENTRATION

Specify the concentration for each species individually. Select the check box for the **Species** to specify the concentration, and then enter a value or expression in the corresponding field. To use another boundary condition for a specific species, click to clear the check box for the concentration of that species.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Flux

The **Flux** node can be used to specify the total species flux across a boundary. The total flux of species c is defined accordingly:

$$-\mathbf{n} \cdot (c\mathbf{u} - D\nabla c) = N_0$$

where N_0 is an arbitrary user-specified flux expression (SI unit: mol/(m^2·s)). For example, N_0 can represent a flux from or into a much larger surrounding environment, a phase change, or a flux due to chemical reactions.

When diffusion is the only transport mechanism present, the flux condition is extended to include a mass transfer term to describe flux into a surrounding environment:

$$-\mathbf{n} \cdot (-D\nabla c) = N_0 + k_\mathrm{c}(c_\mathrm{b} - c)$$

where k_c is a mass transfer coefficient (SI unit: m/s), and c_b is the concentration (SI unit: mol/m^3) in the surroundings of the modeled system (the bulk concentration). The mass transfer coefficient (to be specified) is often given by boundary-layer theory.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the flux.

INWARD FLUX

Specify the flux of each species individually. Select the check box for the **Species** to specify the **Inward flux** $N_{0,c}$ (SI unit: mol/(m^2·s)), and enter a value or expression in the corresponding field. To use another boundary condition for a specific species, click to clear the check box for the mass fraction of that species.

Use a minus sign appropriately when specifying a flux leaving the system.

Inflow

The **Inflow** node adds a boundary condition for an inflow boundary, where the concentration of all species $c = c_0$ is specified. This condition is similar to the Concentration node, except the concentrations of all species must be specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

CONCENTRATION

For the concentration of each species $c_{0,c}$ (SI unit: mol/m^3) enter a value or expression.

CONSTRAINT SETTINGS

These settings are the same as for the Concentration node.

Outflow

Set the **Outflow** condition at outlets where species are transported out of the model domain by a fluid flow. It is assumed that convection is the dominating transport mechanism across outflow boundaries, and therefore that diffusive transport can be ignored, that is:

$$\mathbf{n} \cdot (-D\nabla c) = 0$$

In models where only diffusion occurs, this boundary condition is not available.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

Symmetry

The **Symmetry** node can be used to represent boundaries where the species concentration is symmetric, that is, where there is no mass flux in the normal direction across the boundary.

This boundary condition is identical to that of the No Flux node, but applies to all species and cannot be applied to individual species.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the symmetry condition.

Flux Discontinuity

The **Flux Discontinuity** node represents a discontinuity in the mass flux across an interior boundary:

$$-\mathbf{n}\cdot(\mathbf{N}_\mathrm{d}-\mathbf{N}_\mathrm{u}) = N_0 \qquad \mathbf{N} = (c\mathbf{u}-D\nabla c)$$

where the value N_0 specifies the jump in flux evaluated at the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the flux discontinuity.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to apply the flux discontinuity. An identity pair has to be created first. Ctrl-click to deselect.

FLUX DISCONTINUITY

Specify the jump in species mass flux.

Use a positive value for increasing flux when going from the downside to the upside of the boundary. The boundary normal points in the direction from the downside to the upside of an interior boundary and can be plotted for visualization.

Select the check boxes for the species to specify a **Flux discontinuity** $N_{0,c}$, (SI unit: mol/(m^2·s)) and enter a value or expression for the mass flux jump in the corresponding field. To use a different boundary condition for a specific species, click to clear the check box for the flux discontinuity of that species.

Periodic Condition

The **Periodic Condition** node can be used to define periodicity or antiperiodicity between two boundaries. The node can be activated on more than two boundaries, in which case the feature tries to identify two separate surfaces that can each consist of several connected boundaries. For more complex geometries it might be necessary to right-click and add the **Destination Selection** subnode. With this subnode the boundaries that constitute the source and destination surfaces can be manually specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

- Periodic Condition and Destination Selection

Periodic Boundary Conditions

Point Mass Source

The **Point Mass Source** feature models mass flow originating from an infinitely small domain around a point.

This feature requires at least one of the following licenses: Batteries and Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

For the Reacting Flow in Porous Media, Diluted Species interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, the **Point Mass Source** node is available in two versions, one for the fluid flow (**Fluid Point Source**) and one for the species (**Species Point Source**).

POINT SELECTION

The **Point Mass Source** feature is available in 3D where it can be added to any point and in 2D axisymmetry where it can be added to points on the symmetry axis.

SPECIES SOURCE

Enter the source strength, $\dot{q}_{\mathrm{l,c}}$, for each species (SI unit: mol/(m·s)). A positive value results in species injection from the point into the computational domain, and a negative value means that the species is removed from the computational domain.

Point sources located on a boundary or on an edge affect the adjacent computational domains. This has the effect, for example, that the physical strength of a point source located in a symmetry plane is twice the given strength.

Mass Sources for Species Transport

Thin Diffusion Barrier

Use the **Thin Diffusion Barrier** boundary condition to model a thin layer through which mass is transported by diffusion only. To set up the node, specify the layer thickness and a diffusion coefficient for each transported species.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to define the barrier.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect. A No Flux node is added by default to the **Thin Diffusion Barrier** pair.

THIN DIFFUSION BARRIER

Enter a **Layer thickness** d_{s} (SI unit: m). The default is 0.005 m (5 mm).

Enter a **Diffusion coefficient** $D_{\mathrm{s,c}}$ (SI unit: m^2/s). The default is 0 m^2/s.

Thin Impermeable Barrier

Use the **Thin Impermeable Barrier** feature node to model a thin mass transfer barrier. The feature is available on interior boundaries and introduces a discontinuity in the concentration across the boundary. On each side of the boundary a no flux condition is prescribed for the mass transport. The **Thin Impermeable Barrier** boundary feature can be used to avoid meshing thin structures.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to define the barrier.

13

Fluid Flow

This chapter explains how to use the Laminar Flow interface, found under the **Fluid Flow>Single-Phase Flow** branch (≋) when adding a physics interface, to model and simulate fluid mechanics for laminar, incompressible fluids. The engineering community often uses the term *CFD*, *computational fluid dynamics*, to refer to the numerical simulation of fluids.

Theory of Laminar Flow

The **Single-Phase Flow, Laminar Flow** theory is described in this section:

- General Single-Phase Flow Theory
- Compressible Flow
- The Mach Number Limit
- Incompressible Flow
- The Reynolds Number
- The Boussinesq Approximation
- Theory for the Wall Boundary Condition
- Prescribing Inlet and Outlet Conditions
- Pressure, No Viscous Stress Boundary Condition
- Normal Stress Boundary Condition
- Mass Sources for Fluid Flow
- Numerical Stability—Stabilization Techniques for Fluid Flow
- Solvers for Laminar Flow
- Pseudo Time Stepping for Laminar Flow Models
- Discontinuous Galerkin Formulation
- Particle Tracing in Fluid Flow
- References for the Single-Phase Flow, Laminar Flow Interfaces

The theory about most boundary conditions is found in Ref. 2.

The Laminar Flow interface has many shared features that are only available with some modules. For the theory relating to the advanced features, see the individual module documentation.

General Single-Phase Flow Theory

The single-phase fluid-flow interfaces are based on the Navier-Stokes equations, which in their most general form read

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{13-1}$$

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I} + \tau] + \mathbf{F} \tag{13-2}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T\right) = -(\nabla\cdot\mathbf{q}) + \tau:\mathbf{S} - \frac{T}{\rho}\frac{\partial\rho}{\partial T}\bigg|_p\left(\frac{\partial p}{\partial t} + (\mathbf{u}\cdot\nabla)p\right) + Q \tag{13-3}$$

where

- ρ is the density (SI unit: kg/m^3)
- $\mathbf{u}$ is the velocity vector (SI unit: m/s)
- p is pressure (SI unit: Pa)
- τ is the viscous stress tensor (SI unit: Pa)
- $\mathbf{F}$ is the volume force vector (SI unit: N/m^3)
- C_p is the specific heat capacity at constant pressure (SI unit: J/(kg·K))
- T is the absolute temperature (SI unit: K)
- $\mathbf{q}$ is the heat flux vector (SI unit: W/m^2)
- Q contains the heat sources (SI unit: W/m^3)
- $\mathbf{S}$ is the strain-rate tensor:

$$\mathbf{S} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$$

The operation ":" denotes a contraction between tensors defined by

$$\mathbf{a}:\mathbf{b} = \sum_n \sum_m a_{nm} b_{nm} \tag{13-4}$$

This is sometimes referred to as the double dot product.

Equation 13-1 is the continuity equation and represents conservation of mass. Equation 13-2 is a vector equation which represents conservation of momentum. Equation 13-3 describes the conservation of energy, formulated in terms of temperature. This is an intuitive formulation that facilitates boundary condition specifications.

To close the equation system, Equation 13-1 through Equation 13-3, constitutive relations are needed. For a Newtonian fluid, which has a linear relationship between stress and strain, Stokes (Ref. 1) deduced the following expression:

$$\tau = 2\mu\mathbf{S} - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I} \tag{13-5}$$

The dynamic viscosity, μ (SI unit: Pa·s), for a Newtonian fluid is allowed to depend on the thermodynamic state but not on the velocity field. All gases and many liquids can be considered Newtonian. Examples of non-Newtonian fluids are honey, mud, blood, liquid metals, and most polymer solutions.

When you have the CFD Module or the Microfluidics Module, you can model flows of non-Newtonian fluids using the predefined power law and Carreau models, which describes the dynamic viscosity for non-Newtonian fluids.

Other commonly used constitutive relations are Fourier's law of heat conduction and the ideal gas law.

In theory, the same equations describe both laminar and turbulent flows. In practice, however, the mesh resolution required to simulate turbulence with the **Laminar Flow** interface makes such an approach impractical.

There are several books where derivations of the Navier-Stokes equations and detailed explanations of concepts such as Newtonian fluids can be found. See, for example, the classical text by Batchelor (Ref. 3) and the more recent work by Panton (Ref. 4).

Many applications describe isothermal flows for which Equation 13-3 is decoupled from Equation 13-1 and Equation 13-2.

2D AXISYMMETRIC FORMULATIONS

A 2D axisymmetric formulation of Equation 13-1 and Equation 13-2 requires $\partial/\partial\phi$ to be zero. That is, there must be no gradients in the azimuthal direction. A common additional assumption is however that u_ϕ*=0*. In such cases, the ϕ -equation can be removed from Equation 13-2. The resulting system of equations is both easier to converge and computationally less expensive compared to retaining the ϕ -equation. The default 2D axisymmetric formulation of Equation 13-1 and Equation 13-2 therefore assumes that

$$\partial/\partial\phi = 0$$
$$u_\phi = 0$$

Compressible Flow

The Navier-Stokes equations solved by default in all **Single-Phase Flow** interfaces apply the compressible formulation of the continuity equation:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \qquad (13\text{-}6)$$

and the momentum equation:

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right) + \mathbf{F} \qquad (13\text{-}7)$$

These equations are applicable for incompressible as well as compressible flow with density variations.

The Mach Number Limit

An important dimensionless number in fluid dynamics is the Mach number, Ma, defined by

$$\mathrm{Ma} = \frac{|\mathbf{u}|}{a}$$

where a is the speed of sound. A flow is formally incompressible when $\mathrm{Ma} = 0$. This is theoretically achieved by letting the speed of sound tend to infinity. The Navier-Stokes equations then have the mathematical property that pressure disturbances are instantaneously propagated throughout the entire domain. This results in a parabolic equation system.

The momentum equation, Equation 13-7, is parabolic for unsteady flow and elliptic for steady flow, whereas the continuity equation, Equation 13-6, is hyperbolic for both steady and unsteady flow. The combined system of equations is thus hybrid parabolic-hyperbolic for unsteady flow and hybrid elliptic-hyperbolic for steady flow. An exception occurs when the viscous term in Equation 13-7 becomes vanishingly small, such as at an outflow boundary, in which case the momentum equation becomes locally hyperbolic. The number of boundary conditions to apply on the boundary then depends on the number of characteristics propagating into the computational domain. For the purely hyperbolic system, the number of characteristics propagating from the boundary into the domain changes as the Mach number passes through unity. Hence, the number of boundary conditions required to obtain a numerically well posed system must also change. The compressible formulation of the laminar and turbulent interfaces uses the same boundary conditions as the incompressible formulation, which implies that the compressible interfaces are not suitable for flows with Mach number larger than or equal to one.

The practical Mach number limit is lower than one, however. The main reason is that numerical scheme (stabilization and boundary conditions) of the Laminar Flow interface does not recognize the direction and speed of pressure waves. The fully compressible Navier-Stokes equations do, for example, start to display very sharp gradients already at moderate Mach numbers. But the stabilization in the **Single-Phase Flow** interface does not necessarily capture these gradients. It is impossible to give an exact limit where the low Mach number regime ends and the moderate Mach number regime begins, but a rule of thumb is that the Mach number effects start to appear at $\mathrm{Ma} = 0.3$. For this reason the compressible formulation is referred to as *compressible flow (Ma<0.3)* in COMSOL Multiphysics.

Incompressible Flow

When the temperature variations in a flow are small, a single-phase fluid can often be assumed incompressible; that is, ρ is constant or nearly constant. This is the case for all liquids under normal conditions and also for gases at low velocities. For constant ρ, Equation 13-6 reduces to

$$\rho\nabla\cdot\mathbf{u} = 0 \tag{13-8}$$

and Equation 13-7 becomes

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F} \tag{13-9}$$

The Reynolds Number

A fundamental characteristic in analyses of fluid flow is the Reynolds number:

$$\mathrm{Re} = \frac{\rho UL}{\mu}$$

where U denotes a velocity scale, and L denotes a representative length. The Reynolds number represents the ratio between inertial and viscous forces. At low Reynolds numbers, viscous forces dominate and tend to damp out all disturbances, which leads to laminar flow. At high Reynolds numbers, the damping in the system is very low giving small disturbances the possibility to grow by nonlinear interactions. If the Reynolds number is high enough, the flow field eventually ends up in a chaotic state called turbulence.

Observe that the Reynolds number can have different meanings depending on the length scale and velocity scale. To be able to compare two Reynolds numbers, they must be based on equivalent length and velocity scales.

The **Fluid Flow** interfaces automatically calculate the local cell Reynolds number $\mathrm{Re}^c = \rho|\mathbf{u}|h/(2\mu)$ using the element length h for L and the magnitude of the velocity vector u for the velocity scale U. This Reynolds number is not related to the character of the flow field, but to the stability of the numerical discretization. The risk for numerical oscillations in the solution increases as Re^c grows. The cell Reynolds number is a predefined quantity available for visualization and evaluation (typically it is available as: `spf.cellRe`).

The Boussinesq Approximation

The Boussinesq approximation is a way to treat certain simple cases of buoyant flows without having to use a compressible formulation of the Navier-Stokes equations.

The Boussinesq approximation assumes that variations in density have no effect on the flow field except that they give rise to buoyancy forces. The density is assigned a reference value, ρ_0, everywhere except in the body force term, which is set to

$$\mathbf{F} = (\rho_0 + \Delta\rho)\mathbf{g} \tag{13-10}$$

where $\mathbf{g}$ is the gravity vector. A further simplifications is often possible. Since $\mathbf{g}$ can be written in terms of a potential, Φ, it is possible to write Equation 13-10 as:

$$\mathbf{F} = -\nabla(\rho_0\Phi) + \Delta\rho\mathbf{g}$$

The first part can be canceled out by splitting the true pressure, p, into a hydrodynamic component, P, and a hydrostatic component, $-\rho_0\Phi$. Equation 13-8 and Equation 13-9 are expressed in terms of the hydrodynamic pressure $P = p + \rho_0\Phi$:

$$\rho\nabla\cdot\mathbf{u} = 0 \tag{13-11}$$

$$\rho_0\frac{\partial\mathbf{u}}{\partial t} + (\rho_0\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla P + \nabla\cdot(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}})) + \mathbf{g}\Delta\rho \tag{13-12}$$

To obtain the Boussinesq approximation in this form, enter the expression for $\mathbf{g}\Delta\rho$ for the **Volume Force** feature.

In practice, the shift from p to P can be ignored except where the pressure appears in boundary conditions. The pressure that is specified at boundaries is the hydrodynamic pressure in this case. For example, on a vertical outflow or inflow boundary, the hydrodynamic pressure is typically a constant, while the true pressure is a function of the vertical coordinate.

The system that Equation 13-11 and Equation 13-12 form has its limitations. The main assumption is that the density fluctuations must be small; that is, $\Delta\rho/\rho_0 << 1$. There are also some more subtle constraints that, for example, make the Boussinesq approximation unsuitable for systems of very large dimensions. An excellent discussion of the Boussinesq approximation and its limitations appears in Chapter 14 of Ref. 10.

See Volume Force for the node settings.

Theory for the Wall Boundary Condition

See Wall for the node settings. Note that some modules have additional theory sections describing options available with that module.

SLIP

The Slip condition assumes that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this is a reasonable approximation if the important effect of the wall is to prevent fluid from leaving the domain. Mathematically, the constraint can be formulated as:

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad (-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}))\mathbf{n} = \mathbf{0}$$

The no penetration term takes precedence over the Neumann part of the condition and the above expression is therefore equivalent to

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad \mathbf{K}-(\mathbf{K}\cdot\mathbf{n})\mathbf{n} = \mathbf{0}$$
$$\mathbf{K} = \mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}})\mathbf{n}$$

expressing that there is no flow across the boundary and no viscous stress in the tangential direction.

SLIDING WALL

The Sliding Wall boundary condition is appropriate if the wall behaves like a conveyor belt; that is, the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

In 2D, the tangential direction is unambiguously defined by the direction of the boundary, but the situation becomes more complicated in 3D. For this reason, this boundary condition has slightly different definitions in the different space dimensions.

For 2D and 2D axisymmetric models, the velocity is given as a scalar U_{w} and the condition prescribes

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad \mathbf{u}\cdot\mathbf{t} = U_{\mathrm{w}}$$

where $\mathbf{t} = (n_y, -n_x)$ for 2D and $\mathbf{t} = (n_z, -n_r)$ for axial symmetry.

For 3D models, the velocity is set equal to a given vector $\mathbf{u}_{\mathrm{w}}$ projected onto the boundary plane:

$$\mathbf{u} = \frac{\mathbf{u}_{\mathrm{w}}-(\mathbf{n}\cdot\mathbf{u}_{\mathrm{w}})\mathbf{n}}{\|\mathbf{u}_{\mathrm{w}}-(\mathbf{n}\cdot\mathbf{u}_{\mathrm{w}})\mathbf{n}\|}\|\mathbf{u}_{\mathrm{w}}\|$$

The normalization makes $\mathbf{u}$ have the same magnitude as $\mathbf{u}_{\mathrm{w}}$ even if $\mathbf{u}_{\mathrm{w}}$ is not exactly parallel to the wall.

Prescribing Inlet and Outlet Conditions

The Navier-Stokes equations can show large variations in mathematical behavior, ranging from almost completely elliptic to almost completely hyperbolic. This has implications when it comes to prescribing admissible boundary conditions. There is also a discrepancy between mathematically valid boundary conditions and practically useful boundary conditions.

- Also see Inlet and Outlet for the node settings.
- Pressure, No Viscous Stress Boundary Condition
- Normal Stress Boundary Condition

INLET CONDITIONS

An inlet requires specification of the velocity components. The most robust way to do this is to prescribe a velocity field using a Velocity condition.

A common alternative to prescribing the complete velocity field is to prescribe a pressure, in which case the normal velocity component is specified indirectly via the continuity equation. The pressure can be specified pointwise, which is a mathematically over-constraining but numerically robust formulation. This is achieved using the Pressure, No Viscous Stress condition. Alternatively, the pressure can be specified via a stress condition:

$$-p + 2\mu\frac{\partial u_n}{\partial n} = F_n \tag{13-13}$$

where $\partial u_n/\partial n$ is the normal derivative of the normal velocity component. Equation 13-13 is prescribed by the Normal stress condition. Equation 13-13 is mathematically more stringent compared to specifying the pressure pointwise, can at the same time not guarantee that p obtains the desired value. In practice, p will however be close to F_n, except for low Reynolds number flows where viscous effects are the only effects that balance the pressure.

OUTLET CONDITIONS

The most common approach is to prescribe a pressure via a normal stress condition on the outlet. This is often accompanied by a vanishing tangential stress condition:

$$\mu\frac{\partial u_t}{\partial n} = 0$$

where $\partial u_t/\partial n$ is the normal derivative of the tangential velocity field. It is also possible to prescribe u_t to be zero. The latter option should be used with care since it can have a significant effect on the upstream solution.

The elliptic character of the Navier-Stokes equations mathematically permit specifying a complete velocity field at an outlet. This can however be difficult to apply in practice. The reason being that it is hard to prescribe the outlet velocity so that it at each point is consistent with the interior solution. The adjustment to the specified velocity then occurs across an outlet boundary layer. The thickness of this boundary layer depends on the Reynolds number; the higher the Reynolds number, the thinner the boundary layer.

Pressure, No Viscous Stress Boundary Condition

The Pressure, No Viscous Stress boundary condition specifies vanishing viscous stress along with a Dirichlet condition on the pressure:

$$\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\mathbf{n} = \mathbf{0}, \qquad p = p_0$$

$$\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}, \qquad p = p_0$$

using the compressible and the incompressible formulation, respectively.

This boundary condition physically corresponds to flow entering from a large container. It is numerically stable and admits total control of the pressure level along the entire boundary; however, it can give artifacts on the boundary if the viscous stresses just downstream of the inlet are non-zero. In such situations there are two choices. Either move the boundary farther away to a location where the artifacts do not matter or use another stress type boundary condition present in the Boundary Stress feature.

While the **Pressure, no viscous stress** boundary condition is numerically more robust than the Normal stress condition (which also specifies the pressure), it is also theoretically over-constraining of the flow field (Ref. 4). This theoretical "flaw" is often ignored since it in most cases has no practical implication.

Inlet and Outlet for the node settings. Note that some modules have additional theory sections describing options available with that module.

Normal Stress Boundary Condition

The total stress on the boundary is set equal to a stress vector of magnitude f_0, oriented in the negative normal direction:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = -f_0\mathbf{n}$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

using the compressible and the incompressible formulation, respectively.

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure which for 2D flows is

$$p = 2\mu\frac{\partial u_n}{\partial n}+f_0 \tag{13-14}$$

If $\partial u_n/\partial n$ is small, Equation 13-14 states that $p \approx f_0$.

The Normal Stress condition is the mathematically correct version of the Pressure, No Viscous Stress condition (Ref. 4), but it is numerically less stable.

Pressure Boundary Condition

For single-phase flow, a mathematically correct natural boundary condition for outlets is

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -p_0\mathbf{n} \tag{13-15}$$

This is a normal stress condition together with a no tangential stress condition. When $\mu > 0$ Equation 13-15 can be supplemented with a tangential velocity condition

$$\mathbf{u}\cdot\mathbf{t} = 0 \tag{13-16}$$

If so, the no tangential stress condition is overridden. An issue with Equation 13-15 is that it does not strongly enforce outflow on the boundary. If the prescribed pressure is too high, parts of the outlet can actually have inflow. This is not as much of an issue for the Navier-Stokes equations as it is an issue for scalar transport equations solved along with Navier-Stokes equations. Hence, when applying the **Pressure** boundary condition at an outlet you can further constrain the outflow. With the Suppress backflow option

$$\begin{gathered}(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -\hat{p}_0\mathbf{n}\\ \hat{p}_0 \le p_0\end{gathered} \tag{13-17}$$

the normal stress is adjusted to keep

$$\mathbf{u}\cdot\mathbf{n} \ge 0 \tag{13-18}$$

Equation 13-17 effectively means that the prescribed pressure is p_0 if $\mathbf{u}\cdot\mathbf{n} \geq 0$, but smaller at locations where $\mathbf{u}\cdot\mathbf{n} < 0$. This means that Equation 13-17 does not completely prevent backflow, but the backflow is substantially reduced

Inlet, Outlet, Open Boundary, and No Viscous Stress for the individual node settings. Note that some modules have additional theory sections describing options available with that module.

Mass Sources for Fluid Flow

There are two types of mass sources in the **Single-Phase Flow** interface: point sources and line sources.

These features require at least one of the following licenses: Batteries and Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

POINT SOURCE

A point source is theoretically formed by taking a mass injection/ejection, $\dot{Q}$ (SI unit: kg/(m^3·s)), in a small volume δV and then letting the size of the volume tend to zero while keeping the total mass flux constant. Given a point source strength, $\dot{q}_\mathrm{p}$ (SI unit: kg/s), this can be expressed as

$$\lim_{\delta V \to 0} \int_{\delta V} \dot{Q} = \dot{q}_\mathrm{p} \qquad (13\text{-}19)$$

An alternatively way to form a point source is to assume that mass is injected/extracted through the surface of a small object. Letting the object surface area tend to zero while keeping the mass flux constant, results in the same point source. For this alternative approach, effects resulting from the physical object volume, such as drag and fluid displacement, need to be neglected.

The weak contribution

$$\dot{q}_\mathrm{p}\mathrm{test}(p)$$

is added to a point in the geometry. As can be seen from Equation 13-19, $\dot{Q}$ must tend to plus or minus infinity as δV tends to zero. This means that in theory the pressure also tends to plus or minus infinity.

Observe that “point” refers to the physical representation of the source. A point source can therefore only be added to points in 3D models and to points on the symmetry axis in 2D axisymmetry models. Other geometrical points in 2D models represent physical lines.

The finite element representation of Equation 13-19 corresponds to a finite pressure in a point with the effect of the point source spread out over a region around the point. The size of the region depends on the mesh and on the strength of the source. A finer mesh gives a smaller affected region, but also a more extreme pressure value. It is important not to mesh too finely around a point source since the resulting pressure can result in unphysical values for the density, for example. It can also have a negative effect on the condition number for the equation system.

LINE SOURCE

A line source can theoretically be formed by assuming a source of strength $\dot{Q}$ (SI unit: kg/(m^3·s)), located within a tube with cross-section area δS and then letting δS tend to zero while keeping the total mass flux per unit length constant. Given a line source strength, $\dot{q}_l$ (SI unit: kg/(m·s)), this can be expressed as

$$\lim_{\delta S \to 0} \int_{\delta S} \dot{Q} = \dot{q}_l \tag{13-20}$$

As in the point source case, an alternative approach is to assume that mass is injected/extracted through the surface of a small object. This results in the same mass source, but requires that effects on the fluid resulting from the physical object volume are neglected.

The weak contribution

$$\dot{q}_l \text{test}(p)$$

is added to lines in 3D or to points in 2D (which represent cut-through views of lines). Line sources can also be added to the axisymmetry line in 2D axisymmetry models. It can not, however, be added to geometrical lines in 2D since those represent physical planes.

As with a point source, it is important not to mesh too finely around the line source.

For feature node information, see Line Mass Source and Point Mass Source .

For the **Reacting Flow in Porous Media, Diluted Species** interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, these shared physics nodes are renamed as follows:

- The **Line Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Line Source**) and one for the species (**Species Line Source**).
- The **Point Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Point Source**) and one for the species (**Species Point Source**).

Numerical Stability—Stabilization Techniques for Fluid Flow

The momentum equation (Equation 13-7 or Equation 13-9) is a (nonlinear) convection-diffusion equation. Such equations can easily become unstable if discretized using the Galerkin finite element method. Stabilized finite element methods are usually necessary in order to obtain physical solutions. The stabilization settings are found in the main fluid-flow features. To display this section, click the **Show** button (👁) and select **Stabilization**.

There are three types of stabilization methods available for Navier-Stokes—*streamline diffusion*, *crosswind diffusion*, and *isotropic diffusion*. Streamline diffusion and crosswind diffusion are consistent stabilization methods, whereas isotropic diffusion is an inconsistent stabilization method.

For optimal functionality, the exact weak formulations of and constants in the streamline diffusion and crosswind diffusion methods depend on the order of the shape functions (basis functions) for the elements. The values of constants in the streamline diffusion and crosswind diffusion methods follow Ref. 5 and Ref. 6.

- Numerical Stabilization
- Iterative

STREAMLINE DIFFUSION

For strongly coupled systems of equations, the streamline diffusion method must be applied to the system as a whole rather than to each equation separately. These ideas were first explored by Hughes and Mallet (Ref. 7) and were later extended to Galerkin least-squares (GLS) applied to the Navier-Stokes equations (Ref. 8). This is the streamline diffusion formulation that COMSOL Multiphysics supports. The time-scale tensor is the diagonal tensor presented in Ref. 9.

Streamline diffusion is active by default because it is necessary when convection is dominating the flow.

The governing equations for incompressible flow are subject to the Babuska-Brezzi condition, which states that the shape functions (basis functions) for pressure must be of lower order than the shape functions for velocity. If the incompressible Navier-Stokes equations are stabilized by streamline diffusion, it is possible to use equal-order interpolation. Hence, streamline diffusion is necessary when using first-order elements for both velocity and pressure. This applies also if the model is solved using geometric multigrid (either as a solver or as a preconditioner) and at least one multigrid hierarchy level uses linear Lagrange elements.

CROSSWIND DIFFUSION

Crosswind diffusion can also be formulated for systems of equations, and when applied to the Navier-Stokes equations it becomes a shock-capturing operator. COMSOL Multiphysics supports the formulation in Ref. 8 with a shock capturing viscosity of the Hughes-Mallet type Ref. 7.

Incompressible flows do not contain shock waves, but crosswind diffusion is still useful for introducing extra diffusion in sharp boundary layers and shear layers that otherwise would require a very fine mesh to resolve.

Crosswind diffusion is active by default as it makes it easier to obtain a solution even if the problem is fully resolved by the mesh.

Crosswind diffusion also enables the iterative solvers to use inexpensive presmoothers. If crosswind diffusion is deactivated, more expensive preconditioners must be used instead.

ISOTROPIC DIFFUSION

Isotropic diffusion adds diffusion to the Navier-Stokes equations. Isotropic diffusion significantly reduces the accuracy of the solution but does a very good job at reducing oscillations. The stability of the continuity equation is not improved.

Solvers for Laminar Flow

The Navier-Stokes equations constitute a nonlinear equation system. A nonlinear solver must hence be applied to solve the problem. The nonlinear solver iterates to reach the final solution. In each iteration, a linearized version of the nonlinear system is solved using a linear solver. In the time-dependent case, a time marching method must also be applied. The default suggestions for each of these solver elements are discussed below.

NONLINEAR SOLVER

The nonlinear solver depends if the model solves a stationary or a time-dependent problem.

Stationary Solver

In the stationary case, a fully coupled, damped Newton method is applied. The initial damping factor is low since a full Newton step can be harmful unless the initial values are close to the final solution. The nonlinear solver algorithm automatically regulates the damping factor in order to reach a converged solution.

For advanced models, the automatically damped Newton method might not be robust enough. A pseudo time-stepping algorithm can then be invoked. See Pseudo Time Stepping for Laminar Flow Models.

Time-Dependent Solver

In the time-dependent case, the initial guess for each time step is (loosely speaking) the previous time step, which is a very good initial value for the nonlinear solver. The automatic damping algorithm is then not necessary. The damping factor in the Newton method is instead set to a constant value slightly smaller than one. Also for the same reason, it suffices to update the Jacobian once per time-step.

It is seldom worth the extra computational cost to update the Jacobian more than once per time step. For most models it is more efficient to restrict the maximum time step or possibly lower the damping factor in the Newton method.

Stationary Solver

LINEAR SOLVER

The linearized Navier-Stokes equation system has saddle point character, unless the density depends on the pressure. This means that the Jacobian matrix has zeros on the diagonal. Even when the density depends on the pressure the equation system effectively shares many numerical properties with a saddle point system.

For small 2D models and 3D models, the default solver suggestion is a direct solver. Direct solvers can handle most non-singular systems and are very robust and also very fast for small models. Unfortunately, they become slow for large models and their memory requirement scales as somewhere between $N^{1.5}$and N^2 where N is the number of degrees of freedom in the model. The default suggestion for large 3D models is therefore the iterative GMRES solver. The memory requirement for an iterative solver optimally scales as N.

Geometric Multigrid (GMG) is used to accelerate GMRES. GMG needs smoothers but the saddle point character of the linear system restricts the number of applicable smothers. The choices are further restricted by the anisotropic meshes frequently encountered in fluid-flow problems. Pointwise smoothers, such as SOR, are not very efficient on anisotropic meshes.

The efficiency of the smoothers is highly dependent on the numerical stabilization. Iterative solvers perform at their best when both Streamline Diffusion and Crosswind Diffusion are active.

The default smoother for P1+P1 elements is SCGS. This is an efficient and robust smoother specially designed to solve saddle point systems on meshes that contain anisotropic elements. The SCGS smoother works well even without crosswind diffusion. SCGS can sometimes work for higher order elements, especially if **Method** in the SCGS settings is set to **Mesh element lines**. But there is no guarantee for this, so the default smoother for P2+P1 elements and P3+P2 elements is an SOR Line smoother. SOR Line handles mesh anisotropy, but does not formally address the saddle point character. It does however function in practice provided that streamline diffusion and crosswind diffusion are both active.

A different kind of saddle point character can arise if the equation system contains ODE variables. Some advanced boundary conditionscan add equations with such variables. These variables must be treated with the Vanka algorithm. SCGS includes an option to invoke Vanka. Models with higher order elements must either apply SCGS, or use the Vanka smoother. The latter is the default suggestion for higher order elements, but it does not work optimally for anisotropic meshes.

- Multigrid
- Direct
- Iterative
- SCGS
- SOR Line
- Vanka

TIME-DEPENDENT SOLVER

The default time-dependent solver for Navier-Stokes is the BDF method with maximum order set to two. Higher BDF orders are not stable for transport problems in general nor for Navier-Stokes in particular.

BDF methods have been used for a long time and are known for their stability. However, they can have severe damping effects, especially the lower-order methods. Hence, if robustness is not an issue, a model can benefit from using the generalized-α method instead. Generalized-α is a solver which has properties similar to those of the second-order BDF solver but it is much less diffusive.

Both BDF and generalized-α are per default set to automatically adjust the time step. While this works well for many models, extra efficiency and accuracy can often be gained by specifying a maximum time step. It is also often beneficial to specify an initial time step to make the solver progress smoothly in the beginning of the time series.

Time-Dependent Solver

Pseudo Time Stepping for Laminar Flow Models

A stationary formulation has per definition no time derivatives and Equation 13-9 reduces to:

$$\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]+\mathbf{F} \tag{13-21}$$

Solving Equation 13-21 requires a starting guess that is close enough to the final solution. If no such guess is at hand, the fully transient problem can be solved instead. This is, however, a rather costly approach in terms of computational time. An intermediate approach is to add a fictitious time derivative to Equation 13-21:

$$\rho\frac{\mathbf{u}-\mathrm{nojac}(\mathbf{u})}{\Delta\tilde{t}}+\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]+\mathbf{F}$$

where $\Delta\tilde{t}$ is a *pseudo time step*. Since $\mathbf{u}$–`nojac`($\mathbf{u}$) is always zero, this term does not affect the final solution. It does, however, affect the discrete equation system and effectively transforms a nonlinear iteration into a step of size $\Delta\tilde{t}$ of a time-dependent solver.

Pseudo time stepping is not active per default. The pseudo time step $\Delta\tilde{t}$ can be chosen individually for each element based on the local CFL number:

$$\Delta\tilde{t} = \mathrm{CFL}_{\mathrm{loc}}\frac{h}{|\mathbf{u}|}$$

where h is the mesh cell size. A small CFL number means a small time step. It is practical to start with a small CFL number and gradually increase it as the solution approaches steady state.

If the automatic expression for $\mathrm{CFL}_{\mathrm{loc}}$ is set to the built-in variable `CFLCMP`. The automatic setting then suggests a PID regulator for the pseudo time step in the default solver. The PID regulator starts with a small CFL number and increases $\mathrm{CFL}_{\mathrm{loc}}$ as the solution comes closer to convergence.

For details about the CFL regulator, see Pseudo Time Stepping.

The default manual expression is

$$\begin{gathered}1.3^{\min(\mathrm{niterCMP},\,9)}+\\ \mathrm{if}(\mathrm{niterCMP}>20,\,9\cdot 1.3^{\min(\mathrm{niterCMP}-20,\,9)},\,0)+\\ \mathrm{if}(\mathrm{niterCMP}>40,\,90\cdot 1.3^{\min(\mathrm{niterCMP}-40,\,9)},\,0)\end{gathered} \tag{13-22}$$

The variable `niterCMP` is the nonlinear iteration number. It is equal to one for the first nonlinear iteration. $\mathrm{CFL}_{\mathrm{loc}}$ starts at 1.3 and increases by 30% each iteration until it reaches $1.3^9 \approx 10.6$. It remains there until iteration number 20 at which it starts to increase until it reaches approximately 106. A final increase after iteration number 40 then takes it to 1060. Equation 13-22 can for some advanced flows increase $\mathrm{CFL}_{\mathrm{loc}}$ too slowly or too quickly. $\mathrm{CFL}_{\mathrm{loc}}$ can then be tuned for the specific application.

Discontinuous Galerkin Formulation

Some boundary conditions are implemented using a discontinuous Galerkin formulation. These boundary conditions include

- Wall – Slip
- Periodic Flow Condition
- Flow Continuity

The formulation used in the fluid-flow interfaces in COMSOL Multiphysics is the Symmetric Interior Penalty Galerkin method (SIPG). The SIPG method can be regarded to satisfy the boundary conditions in an integral sense rather than pointwise. More information on SIPG can be found in Ref. 13.

In particular, the SIPG formulation includes a penalty parameter that must be large enough for the formulation to be coercive. The higher the value, the better the boundary condition is fulfilled, but a too high value results in an ill-conditioned equation system. The penalty parameter in COMSOL is implemented according to Ref. 14.

Particle Tracing in Fluid Flow

The Particle Tracing Module is available to assist with these types of modeling problems.

The model Flow Past a Cylinder (model library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**) demonstrates how to add and set up particle tracing in a plot group using the **Particle Tracing with Mass** node. It uses the predefined Khan-Richardson model for the drag force and neglects gravity and buoyancy forces.

It is possible to model particle tracing with COMSOL Multiphysics provided that the impact of the particles on the flow field is negligible. First compute the flow field, and then, as an analysis step, calculate the motion of the particles. The motion of a particle is defined by Newton's second law

$$m\frac{d^2\mathbf{x}}{dt^2} = \mathbf{F}\left(t, \mathbf{x}, \frac{d\mathbf{x}}{dt}\right)$$

where $\mathbf{x}$ is the position of the particle, m the particle mass, and $\mathbf{F}$ is the sum of all forces acting on the particle. Examples of forces acting on a particle in a fluid are the drag force, the buoyancy force, and the gravity force. The drag force represents the force that a fluid exerts on a particle due to a difference in velocity between the fluid and the particle. It includes the viscous drag, the added mass, and the Basset history term. Several empirical expression have been suggested for the drag force. One of those is the one proposed by Khan and Richardson (Ref. 11). That expression is valid for spherical particles for a wide range of particle Reynolds numbers. The particle Reynolds number is defined as

$$\mathrm{Re}_p = \frac{|\mathbf{u}-\mathbf{u}_p|2r\rho}{\mu}$$

where $\mathbf{u}$ is the velocity of the fluid, $\mathbf{u}_p$ the particle velocity, r the particle radius, ρ the fluid density, and μ the dynamic viscosity of the fluid. The empirical expression for the drag force according to Khan and Richardson is

$$\mathbf{F} = \pi r^2\rho|\mathbf{u}-\mathbf{u}_p|(\mathbf{u}-\mathbf{u}_p)[1.84\mathrm{Re}_p^{-0.31} + 0.293\mathrm{Re}_p^{0.06}]^{3.45}$$

References for the Single-Phase Flow, Laminar Flow Interfaces

1. G.G. Stokes, Trans. Camb. Phil. Soc., 8, 287-305, 1845

2. P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley & Sons, 2000.

3. G.K. Batchelor, *An Introduction To Fluid Dynamics*, Cambridge University Press, 1967.

4. R.L. Panton, *Incompressible Flow*, 2nd ed., John Wiley & Sons, 1996.

5. I. Harari and T.J.R. Hughes, "What are *C* and *h*? Inequalities for the Analysis and Design of Finite Element Methods," *Comp. Meth. Appl. Mech. Engrg*, vol. 97, pp. 157–192, 1992.

6. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, "*YZ*β Discontinuity Capturing for Advection-dominated Processes with Application to Arterial Drug Delivery," *Int.J.Num. Meth. Fluids*, vol. 54, pp. 593–608, 2007.

7. T.J.R. Hughes and M. Mallet, "A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive System," *Comp. Meth. Appl. Mech. Engrg*, vol. 58, pp. 305–328, 1986.

8. G. Hauke and T.J.R. Hughes, "A Unified Approach to Compressible and Incompressible Flows," *Comp. Meth. Appl. Mech. Engrg*, vol. 113, pp. 389–395, 1994.

9. G. Hauke, "Simple Stabilizing Matrices for the Computation of Compressible Flows in Primitive Variables," *Comp. Meth. Appl. Mech. Engrg*, vol. 190, pp. 6881–6893, 2001.

10. D.J. Tritton, *Physical Fluid Dynamics*, 2nd ed., Oxford University Press, 1988.

11. J.M. Coulson and J.F. Richardson, "Particle Technology and Separation Processes," *Chemical Engineering, Volume 2*, Butterworth-Heinemann, 2002.

12. J.L. Guermond, P. Minev, and J. Shen, "An overview of projection methods for incompressible flows," *Comp. Meth. Appl. Mech. Engrg*, vol. 195, pp. 6011–6045, 2006.

13. B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*, SIAM, 2008.

14. Y. Epshteyn and B. Rivière, "Estimation of penalty parameters for symmetric interior penalty Galerkin methods," *J. Computational and Applied Mathematics*, vol. 206, pp. 843–872, 2007.

15. R.P. Chhabra and J.F. Richardson, *Non-Newtonian Flow and Applied Rheology*, 2nd ed., Elsivier, 2008.

The Single-Phase Flow, Laminar Flow Interface

The Laminar Flow Interface

The **Laminar Flow (spf)** interface ( ), found under the **Single-Phase Flow** branch ( ) when adding a physics interface, is used to compute the velocity and pressure fields for the flow of a single-phase fluid in the laminar flow regime. A flow will remain laminar as long as the Reynolds number is below a certain critical value. At higher Reynolds numbers, disturbances have a tendency to grow and cause transition to turbulence. This critical Reynolds number depends on the model, but a classical example is pipe flow where the critical Reynolds number is known to be approximately 2000.

The physics interface supports incompressible flows and compressible flows at low Mach numbers (typically less than 0.3). It also supports non-Newtonian fluids.

The equations solved by the **Laminar Flow** interface are the Navier-Stokes equations for conservation of momentum and the continuity equation for conservation of mass.

The **Laminar Flow** interface can be used for stationary and time-dependent analyses. Note that for higher Reynolds numbers, a flow becomes inherently time dependent and three-dimensional, and time-dependent studies have to be used.

When this physics interface is added, the following default nodes are also added in the **Model Builder**—**Fluid Properties**, **Wall** (the default boundary condition is **No slip**), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and volume forces. You can also right-click **Laminar Flow** to select physics from the context menu.

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node that is valid on the axial symmetry boundaries only.

- Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow
- Theory of Laminar Flow

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to these variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first physics interface in the component) is `spf`.

PHYSICAL MODEL

By default the physics interface uses the **Compressible flow (Ma<0.3)** formulation of the Navier-Stokes equations. Select **Incompressible flow** to use the incompressible (constant density) formulation.

DEPENDENT VARIABLES

The following dependent variables (fields) are defined for this physics interface—the **Velocity field** $\mathbf{u}$ (SI unit: m/s) and its components, and the **Pressure** p (SI unit: Pa).

If required, edit the field, component, and dependent variable names. Editing the name of a scalar dependent variable changes both its field name and the dependent variable name. If a new field name coincides with the name of another field of the same type, the fields share degrees of freedom and dependent variable names. A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two fields share a common field name.

CONSISTENT STABILIZATION

To display this section, click the **Show** button ( ) and select **Stabilization**.

The consistent stabilization methods applicable to the **Navier-Stokes equations** are **Streamline diffusion** and **Crosswind diffusion**. These check boxes are selected by default. If required, click to clear one or both of the **Streamline diffusion** and **Crosswind diffusion** check boxes. Observe that using P1+P1 elements requires **Streamline diffusion** to be active. If you deactivate **Streamline diffusion**, make sure that your model uses P2+P1 elements or higher.

INCONSISTENT STABILIZATION

To display this section, click the **Show** button ( ) and select **Stabilization**. By default, the **Isotropic diffusion** check box is not selected for the **Navier-Stokes equations**. Click to select as required.

ADVANCED SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed.

Select the **Use pseudo time stepping for stationary equation form** check box to add pseudo time derivatives to the equation when the **Stationary equation** form is used. When selected, also choose a **CFL number expression**—**Automatic** (the default) or **Manual**. **Automatic** sets the local CFL number (from the Courant–Friedrichs–Lewy condition) to the built-in variable `CFLCMP` which in turn triggers a PID regulator for the CFL number. If **Manual** is selected, enter a **Local CFL number** $\mathrm{CFL}_{\mathrm{loc}}$ (dimensionless).

- Pseudo Time Stepping for Laminar Flow Models
- Pseudo Time Stepping

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. It controls the discretization (the element types used in the finite element formulation). From the **Discretization of fluids** list select the element order for the velocity components and the pressure: **P1+P1** (the default), **P2+P1**, or **P3+P2**.

- **P1+P1** (the default) means linear elements for both the velocity components and the pressure field. This is the default element order for the **Laminar Flow** interface. Linear elements are computationally cheaper than higher-order elements and are also less prone to introducing spurious oscillations, thereby improving the numerical robustness. P1+P1 elements require streamline diffusion to be a numerically valid discretization. Make sure that **Streamline Diffusion** in the Consistent Stabilization section is selected when using P1+P1 elements.
- **P2+P1** means second-order elements for the velocity components and linear elements for the pressure field.
- **P3+P2** means third-order elements for the velocity components and second-order elements for the pressure field. This can add additional accuracy but it also adds additional degrees of freedom compared to P2+P1 elements.

Specify the **Value type when using splitting of complex variables**—**Real** (the default) or **Complex** for each of the variables in the table.

- Show More Physics Options
- Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow
- Theory of Laminar Flow

Flow Past a Cylinder: model library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**

Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow

The following nodes are for all physics interfaces found under the **Fluid Flow>Single-Phase Flow** branch (≋) when adding a physics interface. Other physics interfaces also share many of these domain, boundary, pair, and point nodes.

These nodes, listed in alphabetical order, are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

- No Viscous Stress
- Flow Continuity
- Fluid Properties
- Initial Values
- Inlet
- Line Mass Source*
- Open Boundary
- Outlet
- Periodic Flow Condition
- Point Mass Source*
- Pressure Point Constraint
- Symmetry
- Volume Force
- Wall

* A feature that might require an additional license

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and adds an **Axial Symmetry** node that is valid on the axial symmetry boundaries only.

Fluid Properties

The **Fluid Properties** node adds the momentum equations solved by the physics interface, except for volume forces which are added by the Volume Force feature. The node also provides an interface for defining the material properties of the fluid.

MODEL INPUTS

Edit input variables to the fluid-flow equations if required. For fluid flow, these are typically introduced when a material requiring inputs has been selected.

Absolute Pressure

This input appears when a material requires the absolute pressure as model input. The absolute pressure input controls the pressure used to evaluate material properties, but it also relates to the value of the pressure field. There are usually two ways to calculate the pressure when describing fluid flow. Either solve for the absolute pressure or for a pressure (often denoted gauge pressure) that relates to the absolute pressure through a reference pressure.

Which option to choose usually depends on the system and the equations being solved. For example, in a unidirectional incompressible flow problem, the pressure drop over the modeled domain is probably many orders of magnitude smaller than the atmospheric pressure, which, when included, reduces the stability and convergence properties of the solver. In other cases, you can solve for the absolute pressure, such as when the pressure is part of an expression for the gas volume or the diffusion coefficients.

The default **Absolute pressure** p_A (SI unit: Pa) is p+p_{ref} where p defaults to the pressure variable from the Navier-Stokes or RANS equations and p_{ref} to `1[atm]` (1 atmosphere = 101,325 Pa). The default setting is to solve for a gauge pressure.

If the pressure field instead is an absolute pressure field, clear the **Reference pressure** check box.

To model an incompressible fluid, set **Absolute pressure** p_A to **User defined** and enter the desired pressure level in the edit field. The default value is `1[atm]`.

This makes it possible to use a system-based (gauge) pressure as the pressure variable while automatically including the reference pressure in places where it is required, such as for gas flow governed by the gas law. While this check box maintains control over the pressure variable and instances when absolute pressure is required within this specific physics interface, it can not do so within other physics interfaces that it is coupled to. In such models, check the coupling between any physics interfaces using the same variable.

FLUID PROPERTIES

The default **Density** ρ (SI unit: kg/m^3) uses the value **From material**. Select **User defined** to enter a different value or expression.

The default **Dynamic viscosity** μ (SI unit: Pa·s) uses the value **From material** and describes the relationship between the shear rate and the shear stresses in a fluid. Intuitively, water and air have low viscosities, and substances often described as thick (such as oil) have higher viscosities. Select **User defined** to define a different value or expression.

Using the built-in variable for the shear rate magnitude, `spf.sr`, makes it possible to define arbitrary expressions of the dynamic viscosity as a function of the shear rate.

Volume Force

The **Volume Force** node specifies the volume force **F** on the right-hand side of the momentum equation. Use it, for example, to incorporate the effects of gravity in a model.

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot\left[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right] + \mathbf{F}$$

If several volume force nodes are added to the same domain, then the sum of all contributions are added to the momentum equations.

VOLUME FORCE

Enter the components of the **Volume force** $\mathbf{F}$ (SI unit: N/m^3). The defaults for all components are 0 N/m^3.

The Boussinesq Approximation

Initial Values

The **Initial Values** node adds initial values for the velocity field and the pressure that can serve as initial conditions for a transient simulation or as an initial guess for a nonlinear solver.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the component includes.

INITIAL VALUES

Enter values or expressions for the initial value of the **Velocity field** $\mathbf{u}$ (SI unit: m/s) and the **Pressure** p (SI unit: Pa). The default values are 0 m/s and 0 Pa, respectively.

Wall

The **Wall** node includes a set of boundary conditions describing the fluid-flow condition at a wall.

- No Slip (the default)
- Slip
- Sliding Wall
- Moving Wall
- Leaking Wall

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and can not be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

BOUNDARY CONDITION

Select a **Boundary condition** for the wall.

No Slip

No slip is the default boundary condition for a stationary solid wall for laminar flow. The condition prescribes $\mathbf{u}$ = 0, that is, the fluid at the wall is not moving.

Slip

The **Slip** option prescribes a no-penetration condition, $\mathbf{u}\cdot\mathbf{n}=0$. It hence implicitly assumes that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this can be a reasonable approximation if the main effect of the wall is to prevent fluid from leaving the domain.

Sliding Wall

The **Sliding wall** boundary condition is appropriate if the wall behaves like a conveyor belt; that is, the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

For 3D models, enter the components of the **Velocity of the sliding wall** $\mathbf{u}_\mathrm{w}$ (SI unit: m/s). If the velocity vector entered is not in the plane of the wall, COMSOL Multiphysics projects it onto the tangential direction. Its magnitude is adjusted to be the same as the magnitude of the vector entered.

For 2D models, the tangential direction is unambiguously defined by the direction of the boundary, but the situation becomes more complicated in 3D. For this reason, the sliding wall boundary condition has slightly different definitions in the different space dimensions. Enter the components of the **Velocity of the tangentially moving wall** U_w (SI unit: m/s).

Moving Wall

If the wall moves, so must the adjacent fluid. Hence, this boundary condition prescribes $\mathbf{u} = \mathbf{u}_\mathrm{w}$. Enter the components of the **Velocity of moving wall** $\mathbf{u}_\mathrm{w}$ (SI unit: m/s).

Specifying this boundary condition does not automatically cause the associated wall to move. An additional **Moving Mesh** interface needs to be added to physically track the wall movement in the spatial reference frame.

Leaking Wall

Use this boundary condition to simulate a wall where fluid is leaking into or leaving through a perforated wall $\mathbf{u} = \mathbf{u}_\mathrm{l}$. Enter the components of the **Fluid velocity** $\mathbf{u}_\mathrm{l}$ (SI unit: m/s).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**.

For the No Slip, Moving Wall, and Leaking Wall boundary conditions, select an option from the **Apply reaction terms on:** list—**All physics (symmetric)** or **Individual dependent variables**. The other types of wall boundary conditions with constraints use **Individual dependent variables** constraints only.

Select the **Use weak constraints** check box (not available for the Sliding Wall condition) to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

- Theory for the Wall Boundary Condition
- Moving Mesh Interface

Inlet

The **Inlet** node includes a set of boundary conditions describing the fluid-flow conditions at an inlet. The **Velocity** boundary condition is the default.

In many cases the **Inlet** boundary conditions are available, some of them slightly modified, for the Outlet type as well. For certain conditions there is nothing in the mathematical formulations to prevent the fluid from leaving the domain through boundaries where the **Inlet** type is specified.

- Prescribing Inlet and Outlet Conditions
- Pressure, No Viscous Stress Boundary Condition
- Normal Stress Boundary Condition

BOUNDARY CONDITION

Select a **Boundary condition** for the inlet—Velocity (the default), Pressure, No Viscous Stress, or Normal Stress.

After selecting a **Boundary Condition** from the list, a section with the same name displays underneath. For example, if **Velocity** is selected, a **Velocity** section displays where further settings are defined for the velocity.

VELOCITY

The **Velocity** boundary condition is available for the Inlet and Outlet boundary nodes.

- Select **Normal inflow velocity** (the default) to specify a normal inflow velocity magnitude $\mathbf{u} = -\mathbf{n}\mathrm{U}_0$ where $\mathbf{n}$ is the boundary normal pointing out of the domain. Enter the velocity magnitude U_0 (SI unit: m/s). The default is 0 m/s.
- If **Velocity field** is selected, it sets the velocity equal to a given velocity vector $\mathbf{u}_0$ when $\mathbf{u} = \mathbf{u}_{0.}$ Enter the components of $\mathbf{u}_0$ (SI unit: m/s). The defaults are 0 m/s.

PRESSURE, NO VISCOUS STRESS

The **Pressure, no viscous stress** boundary condition is available for the Inlet boundary node. It specifies vanishing viscous stress along with a Dirichlet condition on the pressure. Enter the **Pressure** p_0 (SI unit: Pa) at the boundary. The default is 0 Pa.

Depending on the pressure field in the rest of the domain, an inlet boundary with this condition can become an outlet boundary.

NORMAL STRESS

The **Normal stress** boundary condition is available for the Inlet, Outlet (via the **Pressure** condition), Open Boundary, and No Viscous Stress nodes. Enter the magnitude of **Normal stress** f_0 (SI unit: N/m^2). This implicitly imposes $p \approx f_0$. The default is 0 N/m^2.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

When Velocity or Pressure, No Viscous Stress are selected as the **Boundary condition**, and to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required.

Outlet

The **Outlet** node includes a set of boundary conditions describing fluid-flow conditions at an outlet. Pressure is the default. Other options are based on individual licenses. Selecting appropriate outlet conditions for the Navier-Stokes equations is not a trivial task. Generally, if there is something interesting happening at an outflow boundary, extend the computational domain to include this phenomenon.

Some of the formulations for the **Outlet** type are also available, possibly slightly modified, in other boundary types. For certain conditions there is nothing in the mathematical formulations to prevent the fluid from entering a domain through boundaries where the **Outlet** boundary type is specified.

Prescribing Inlet and Outlet Conditions

BOUNDARY CONDITION

Select a **Boundary condition** for the outlet—Pressure (the default), or Velocity.

The **Velocity** boundary condition is described for the **Inlet** node.

Pressure

The Pressure condition specifies the normal stress which in most cases is approximately equal to the pressure. The tangential stress component is set to 0 N/m2.

- Enter the **Pressure** p_0 (SI unit: Pa) at the boundary. The default is 0 Pa.
- Select the **Normal flow** check box to change the no tangential stress condition to a no tangential velocity condition. This forces the flow to exit (or enter) the domain perpendicularly to the outlet boundary.
- The **Suppress backflow** check box is selected by default. This option adjusts the outlet pressure in order to prevent fluid from entering the domain through the boundary.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

When Velocity or Pressure is selected as the **Boundary condition**, and to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required.

Symmetry

The **Symmetry** node adds a boundary condition that describes symmetry boundaries in a fluid-flow simulation. The boundary condition for symmetry boundaries prescribes no penetration and vanishing shear stresses. The boundary condition is a combination of a Dirichlet condition and a Neumann condition:

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad \left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = \mathbf{0}$$

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad (-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}}))\mathbf{n} = \mathbf{0}$$

for the compressible and incompressible formulations. The Dirichlet condition takes precedence over the Neumann condition, and the above equations are equivalent to the following equation for both the compressible and incompressible formulations:

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad \mathbf{K}-(\mathbf{K}\cdot\mathbf{n})\mathbf{n} = \mathbf{0}$$
$$\mathbf{K} = \mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}})\mathbf{n}$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

For 2D axial symmetry, a boundary condition does not need to be defined. For the symmetry axis at $r = 0$, the software automatically provides a condition that prescribes $u_r = 0$ and vanishing stresses in the z direction and adds an **Axial Symmetry** node that implements this condition on the axial symmetry boundaries only.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

Open Boundary

The **Open Boundary** node adds boundary conditions describing boundaries in contact with large volumes of fluid. Fluid can both enter and leave the domain on boundaries with this type of condition.

BOUNDARY CONDITIONS

Select a **Boundary condition** for the open boundaries—Normal Stress (the default) or Constraint Settings. The **Normal stress** condition is described for the **Inlet** node.

No Viscous Stress

The **No Viscous Stress** condition specifies vanishing viscous stress on the outlet. This condition does not provide sufficient information to fully specify the flow at the outlet and must at least be combined with pressure constraints on adjacent points.

If **No viscous stress** is selected, it prescribes vanishing viscous stress:

$$\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\mathbf{n} = \mathbf{0}$$

$$\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

using the compressible and the incompressible formulations.

This condition can be useful in some situations because it does not impose any constraint on the pressure. A typical example is a model with volume forces that give rise to pressure gradients that are hard to prescribe in advance. To make the model numerically stable, combine this boundary condition with a point constraint on the pressure.

Boundary Stress

The **Boundary Stress** node adds a boundary condition that represents a very general class of conditions also known as traction boundary conditions.

BOUNDARY CONDITION

Select a **Boundary condition** for the boundary stress—**General stress** (the default), Normal Stress (described for the Inlet node), or **Normal stress, normal flow**.

General Stress

When **General stress** is selected, enter the components for the **Stress** $\mathbf{F}$ (SI unit: N/m^2).The total stress on the boundary is set equal to a given stress $\mathbf{F}$:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = \mathbf{F}$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = \mathbf{F}$$

using the compressible and the incompressible formulations.

This boundary condition implicitly sets a constraint on the pressure that for 2D flows is

$$p = 2\mu\frac{\partial u_n}{\partial n}-\mathbf{n}\cdot\mathbf{F} \tag{13-23}$$

If $\partial u_n/\partial n$ is small, Equation 13-23 states that $p \approx -\mathbf{n}\cdot\mathbf{F}$.

Normal Stress, Normal Flow

If **Normal stress, normal flow** is selected, enter the magnitude of the **Normal stress** f_0 (SI unit: N/m^2).

In addition to the stress condition set in the Normal Stress condition, the **Normal stress, normal flow** condition also prescribes that there must be no tangential velocities on the boundary:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = -\mathrm{f}_0\mathbf{n}, \qquad \mathbf{t}\cdot\mathbf{u} = 0$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -\mathrm{f}_0\mathbf{n}, \qquad \mathbf{t}\cdot\mathbf{u} = 0$$

using the compressible and the incompressible formulations.

This boundary condition also implicitly sets a constraint on the pressure that for 2D flows is

$$p = 2\mu\frac{\partial u_n}{\partial n}+\mathrm{f}_0 \tag{13-24}$$

If $\partial u_n/\partial n$ is small, Equation 13-24 states that $p \approx \mathrm{f}_0$.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

If Normal Stress, Normal Flow is selected as the **Boundary condition**, then to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required.

Periodic Flow Condition

The **Periodic Flow Condition** splits its selection into a source group and a destination group. Fluid that leaves the domain through one of the destination boundaries enters the domain through the corresponding source boundary. This corresponds to a situation where the geometry is a periodic part of a larger geometry. If the boundaries are not parallel to each other, the velocity vector is automatically transformed.

If the boundaries are curved, it is recommended to only include two boundaries.

No input is required when **Compressible flow (Ma<0.3)** is selected as the **Compressibility** option under the **Physical Model** section for the physics interface. Typically when a periodic boundary condition is used with a compressible flow the pressure is the same at both boundaries and the flow is driven by a volume force.

PRESSURE DIFFERENCE

This section is available when **Incompressible flow** is selected as the **Compressibility** option under the **Physical Model** section for the physics interface.

Enter a value or expression for the pressure difference, $p_{\mathrm{src}} - p_{\mathrm{dst}}$ (SI unit: Pa). This pressure difference can, for example, drive the fully developed flow in a channel. The default is 0 Pa.

To set up a periodic boundary condition select both boundaries in the **Periodic Flow Condition** node. COMSOL Multiphysics automatically assigns one boundary as the source and the other as the destination. To manually set the destination selection, add a **Destination Selection** node to the **Periodic Flow Condition** node. All destination sides must be connected.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

Periodic Boundary Conditions

Flow Continuity

The **Flow Continuity** node is suitable for pairs where the boundaries match; it prescribes that the flow field is continuous across the pair.

A Wall subnode is added by default and it applies to the parts of the pair boundaries where a source boundary lacks a corresponding destination boundary and vice versa. The **Wall** feature can be overridden by any other boundary condition that applies to exterior boundaries. Right-click the **Flow Continuity** node to add additional subnodes.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

Point Mass Source

The **Point Mass Source** feature models mass flow originating from an infinitely small domain centered around a point.

For the Reacting Flow in Porous Media, Diluted Species interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, the **Point Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Point Source**) and one for the species (**Species Point Source**).

This feature requires at least one of the following licenses: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

POINT SELECTION

The **Point Mass Source** feature is available in 3D where it can be added to any point and in 2D axisymmetry where it can be added to points on the symmetry axis.

SOURCE STRENGTH

Enter the **Mass flux**, $\dot{q}_\mathrm{p}$, for the source (SI unit: kg/s). A positive value results in mass being ejected from the point into the computational domain. A negative value results in mass being removed from the computational domain.

Point sources located on a boundary or on an edge affects the adjacent computational domains. This has the effect, for example, that the physical strength of a point source located in a symmetry plane is twice the given strength.

Mass Sources for Fluid Flow

Line Mass Source

The **Line Mass Source** feature models mass flow originating from a tube region with infinitely small radius.

> For the Reacting Flow in Porous Media, Diluted Species interface, which is available with the CFD Module, Chemical Reaction Engineering Module, or Batteries & Fuel Cells Module, the **Line Mass Source** node is available as two nodes, one for the fluid flow (**Fluid Line Source**) and one for the species (**Species Line Source**).

> This feature requires at least one of the following licenses: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Microfluidics Module, Pipe Flow Module, or Subsurface Flow Module.

SELECTION

The **Line Mass Source** feature is available for all dimensions, but the applicable selection differs between the dimensions.

MODEL DIMENSION	APPLICABLE GEOMETRICAL ENTITY
2D	Points
2D Axisymmetry	Points not on the symmetry axis and the symmetry axis
3D	Edges

SOURCE STRENGTH

Enter the **Mass flux**, $\dot{q}_l$, for the source (SI unit: kg/(s·m)). A positive value results in mass being ejected from the line into the computational domain and a negative value means that mass is removed from the computational domain.

Line sources located on a boundary affect the adjacent computational domains. This, for example, has the effect that the physical strength of a line source located in a symmetry plane is twice the given strength.

> Mass Sources for Fluid Flow

Pressure Point Constraint

The **Pressure Point Constraint** node adds a pressure constraint at a point. If it is not possible to specify the pressure level using a boundary condition, the pressure level must be set in some other way, for example, by specifying a fixed pressure at a point.

PRESSURE CONSTRAINT

Enter a point constraint for the **Pressure** p_0 (SI unit: Pa). The default is 0 Pa.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

14

Heat Transfer Modeling

This chapter describes the different types of Heat Transfer interfaces (Heat Transfer in Solids and Heat Transfer in Fluids), and the Joule Heating interface, all found under the **Heat Transfer** branch ( ) when adding a physics interface.

Heat Transfer Theory

The Heat Transfer Interfacetheory is described in this section. This section reviews the theory about the heat transfer equations in COMSOL Multiphysics and heat transfer in general. For more detailed discussions of the fundamentals of heat transfer, see Ref. 1 and Ref. 3.

What is Heat Transfer?

Heat transfer is defined as the movement of energy due to a difference in temperature. It is characterized by the following mechanisms:

- *Conduction*—Heat conduction occurs as a consequence of different mechanisms in different media. Theoretically it takes place in a gas through collisions of molecules; in a fluid through oscillations of each molecule in a "cage" formed by its nearest neighbors; in metals mainly by electrons carrying heat and in other solids by molecular motion which in crystals take the form of lattice vibrations known as phonons. Typical for heat conduction is that the heat flux is proportional to the temperature gradient.
- *Convection*—Heat convection (sometimes called heat advection) takes place through the net displacement of a fluid which transports the heat content with its velocity. The term convection (especially convective cooling and convective heating) also refers to the heat dissipation from a solid surface to a fluid, typically described by a heat transfer coefficient.
- *Radiation*—Heat transfer by radiation takes place through the transport of photons. Participating (or semitransparent) media absorb, emit and scatter photons. Opaque surfaces absorb or reflect them.

The Heat Equation

The fundamental law governing all heat transfer is the first law of thermodynamics, commonly referred to as the principle of conservation of energy. However, internal energy, U, is a rather inconvenient quantity to measure and use in simulations. Therefore, the basic law is usually rewritten in terms of the temperature, T. For a fluid, the resulting *heat equation* is:

$$\rho C_p\left(\frac{\partial T}{\partial t}+(\mathbf{u}\cdot\nabla)T\right)=-(\nabla\cdot\mathbf{q})+\tau:\mathbf{S}-\frac{T}{\rho}\frac{\partial\rho}{\partial T}\bigg|_p\left(\frac{\partial p}{\partial t}+(\mathbf{u}\cdot\nabla)p\right)+Q \tag{14-1}$$

where

- ρ is the density (SI unit: kg/m^3)
- C_p is the specific heat capacity at constant pressure (SI unit: J/(kg·K))
- T is the absolute temperature (SI unit: K)
- $\mathbf{u}$ is the velocity vector (SI unit: m/s)
- $\mathbf{q}$ is the heat flux by conduction (SI unit: W/m^2)
- p is the pressure (SI unit: Pa)
- τ is the viscous stress tensor (SI unit: Pa)
- $\mathbf{S}$ is the strain-rate tensor (SI unit: 1/s):

$$\mathbf{S}=\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)$$

- Q contains heat sources other than viscous heating (SI unit: W/m^3)

For a detailed discussion of the fundamentals of heat transfer, see Ref. 1.

Specific heat capacity at constant pressure is the amount of energy required to raise one unit of mass of a substance by one degree while maintained at constant pressure. This quantity is also commonly referred to as *specific heat* or *specific heat capacity*.

In deriving Equation 14-1, a number of thermodynamic relations have been used. The equation also assumes that mass is always conserved, which means that the density and the velocity must be related through:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v}) = 0$$

The **Heat Transfer** interfaces use Fourier's law of heat conduction, which states that the conductive heat flux, **q**, is proportional to the temperature gradient:

$$q_i = -k\frac{\partial T}{\partial x_i} \tag{14-2}$$

where k is the thermal conductivity (SI unit: W/(m·K)). In a solid, the thermal conductivity can be anisotropic (that is, it has different values in different directions). Then k becomes a tensor

$$k = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix}$$

and the conductive heat flux is given by

$$q_i = -\sum_j k_{ij}\frac{\partial T}{\partial x_j}$$

Fourier's law applies for symmetric thermal conductivity tensors. Non symmetric tensors lead to unphysical results.

The second term on the right-hand side of Equation 14-1 represents viscous heating in the fluid. An analogous term arises from the internal viscous damping of a solid. The operation ":" is a contraction and can in this case be written on the following form:

$$\mathbf{a}:\mathbf{b} = \sum_n\sum_m a_{nm}b_{nm}$$

The third term represents pressure work and is the result of heating under adiabatic compression as well as some thermoacoustic effects. It is generally small for low Mach number flows. A similar term can be included to account for thermoelastic effects in solids.

Inserting Equation 14-2 into Equation 14-1, reordering the terms and ignoring viscous heating and pressure work put the heat equation into a more familiar form:

$$\rho C_p\frac{\partial T}{\partial t}+\rho C_p\mathbf{u}\cdot\nabla T = \nabla\cdot(k\nabla T)+Q$$

The **Heat Transfer in Fluids** interface solves this equation for the temperature, T. If the velocity is set to zero, the equation governing purely conductive heat transfer is obtained:

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot (-k\nabla T) = Q$$

A Note on Heat Flux and Balance

The concept of heat flux is not as simple as it might first appear. The reason is that heat is not a conserved quantity. The conserved quantity is instead the total energy. Hence, there is both a heat flux and an energy flux that are similar but not identical.

This section briefly describes the theory for the variables for Total Energy Flux and Total Heat Flux, used when computing heat balance. The definitions of these postprocessing variables do not affect the computational results, only variables available for results analysis and visualization.

TOTAL ENERGY FLUX

The total energy flux for a fluid is equal to (Ref. 4, chapter 3.5)

$$\rho \mathbf{u}(H_0 + \Psi) - k\nabla T - \tau \cdot \mathbf{u} + q_\mathrm{r} \tag{14-3}$$

Above, H_0 is the total enthalpy

$$H_0 = H + \frac{1}{2}(\mathbf{u} \cdot \mathbf{u})$$

where in turn H is the enthalpy. In Equation 14-3 τ is the viscous stress tensor and q_r is the radiative heat flux. Ψ in Equation 14-3 is the force potential. It has a simple form in some special cases, for example, for gravitational effects (Chapter 1.4 in Ref. 4), but it is in general rather difficult to derive. Potential energy is therefore often excluded and the total energy flux is approximated by

$$\rho \mathbf{u}\left(H + \frac{1}{2}(\mathbf{u} \cdot \mathbf{u})\right) - k\nabla T - \tau \cdot \mathbf{u} + q_\mathrm{r} \tag{14-4}$$

For a simple compressible fluid, the enthalpy, H, has the form (Ref. 5)

$$H = H_\mathrm{ref} + \int_{T_\mathrm{ref}}^{T} C_p dT + \int_{p_\mathrm{ref}}^{p} \frac{1}{\rho}\left(1 + \frac{T}{\rho}\left(\frac{\partial \rho}{\partial T}\right)\bigg|_p\right) dp \tag{14-5}$$

where p is the absolute pressure. The reference enthalpy, H_ref, is the enthalpy at reference temperature, T_ref, and reference pressure, p_ref. T_ref is 298.15 K and p_ref is one atmosphere. In theory, any value can be assigned to H_ref (Ref. 7), and COMSOL Multiphysics sets it to 0 J/kg by default.

The two integrals in Equation 14-5 are sometimes referred to as the *sensible enthalpy* (Ref. 7). These are evaluated by numerical integration. The second integral is only included for gas/liquid since it is commonly much smaller than the first integral for solids and it is identically zero for ideal gases.

For the evaluation of H to work, it is important that the dependencies of C_p, ρ, and γ on the temperature are prescribed either via model inputs or as functions of the temperature variable. If C_p, ρ, or γ depends on the pressure, that dependency must be prescribed either via a model input or by using the variable pA, which is the variable for the absolute pressure in COMSOL Multiphysics.

Total Heat Flux

The total heat flux vector is defined as (Ref. 6):

$$\rho \mathbf{u} E - k\nabla T + q_{\mathrm{r}} \tag{14-6}$$

where E is the internal energy. It is related to the enthalpy via

$$H = E + \frac{p}{\rho} \tag{14-7}$$

Compared to the total energy flux, the total heat flux do not have viscous- and pressure-related terms.

Conservation Equations in Differential Forms

According to the First Law of Thermodynamics, internal energy is the conserved quantity at the microscopic scale:

$$dE = \delta Q - \delta W_{\mathrm{nc}} \tag{14-8}$$

Here, δQ and δW_{nc} are the path-dependent heat and work contributions to the variation of internal energy dE. Some mass and momentum conservation equations are often solved together with the heat equation. They may take the form of the Navier-Stokes equations in Fluid Dynamics, Newton’s Laws of Motion in Solid Mechanics or Maxwell's equations in Electromagnetism for instance, and can be expressed by the following differential form of energy equation:

$$dE_{\mathrm{k}} + dE_{\mathrm{p}} = \delta W_{\mathrm{nc}} \tag{14-9}$$

where dE_{k} and dE_{p} are the kinetic and potential energy variations, respectively. The variation of mechanical work from nonconservative forces, δW_{nc}, is transmitted to the internal energy by the corresponding work, $-\delta W_{\mathrm{nc}}$, as the last term of Equation 14-8. The sources δW_{nc} include, amongst others, viscous heating and pressure work in Fluid Dynamics, Joule heating in Electromagnetism, friction and plastic deformation in Solid Mechanics. By combining Equation 14-8 and Equation 14-9, the complete First Law reads:

$$d(E + E_{\mathrm{k}}) = \delta Q + \delta W \tag{14-10}$$

where δW, equal to $-dE_{\mathrm{p}}$, is the mechanical work from conservative forces.

Heat Balance

This paragraph assumes a heat transfer model that only solves for the temperature T. The velocity field $\mathbf{u}$ and pressure field p are user-defined or computed from another physics interface. In this case, the heat balance in a domain follows the identity below (chapter 11.2 in Ref. 8), derived from Equation 14-8. It expresses the idea that internal energy variations in time and net heat flux are balanced by external heat and work sources.

$$\frac{d}{dt}\int_{\Omega} \rho E d\omega + \int_{\partial\Omega_{\mathrm{ext}}} (\rho \mathbf{u} E - k\nabla T + \mathbf{q}_{\mathrm{r}}) \cdot \mathbf{n} d\sigma = Q_{\mathrm{Int}} - W_{\mathrm{ns,\,Int}} \tag{14-11}$$

The different variables in this formula are defined in Total Heat Flux and Total Energy Flux. For this equality to be true, the provided velocity field $\mathbf{u}$ and pressure field p must satisfy a mass and a momentum conservation equation such as the Navier-Stokes Equations or governing equations of continuum mechanics. The nonconservative work from the Navier-Stokes equations, $W_{\mathrm{ns,Int}}$ (definition in Table 14-1), contains both pressure work and viscous heating. The heat sources Q_{Int} include domain sources, interior boundary, edge and point sources, and radiative source at interior boundaries.

In 2D and 3D models, if isolated point or edge source is not adjacent to a boundary, these are not included in Q_{Int}. In this case, these need to be computed separately.

Four kinds of compensating heat power contributions are thus distinguished and available as COMSOL Multiphysics predefined variables:

- The *total accumulated heat power* (SI unit: W), `dEiInt`,
- the *total net heat power* (SI unit: W), `ntfluxInt`, integral on exterior boundaries of the total heat flux,
- the *total heat source* (SI unit: W), `QInt`,
- the *total fluid losses* (SI unit: W), `WnsInt`.

Table 14-1 summarizes the mathematical definitions of these variables.

TABLE 14-1: GLOBAL POST-PROCESSING VARIABLE FOR TOTAL HEAT BALANCE

VARIABLE NAME	MATHEMATICAL DEFINITION
`dEiInt`	$\frac{d}{dt}\int_{\Omega}\rho E d\omega$
`ntfluxInt`	$\int_{\partial\Omega_{\text{ext}}}(\rho\mathbf{u}E - k\nabla T + \mathbf{q}_{\text{r}})\cdot\mathbf{n}d\sigma$
`QInt`	$\int_{\Omega}Qd\omega + \int_{\partial\Omega_{\text{int}}}Q_{\text{b}}d\omega + \int_{\partial\Omega_{\text{""}}}Q_{\text{r}}d\omega$
`WnsInt`	$\int_{\Omega}(p_{\text{A}}\nabla\cdot\mathbf{u})d\omega + \int_{\Omega}(-\tau:\nabla\mathbf{u})d\omega$

Here, Ω_{ext} and Ω_{int} denote the exterior and interior boundaries, respectively.

According to Equation 14-11, the following equality between COMSOL Multiphysics variables holds:

```
dEiInt + ntfluxInt = QInt - WnsInt
```

This is the most general form that can be used for time-dependent models. At steady-state, the formula is simplified. The accumulated heat power equals zero so the total net heat power, sum of incoming and outgoing powers, should correspond to the heat and work sources:

```
ntfluxInt = QInt - WnsInt
```

The sign convention used in COMSOL Multiphysics for `QInt` is the following: positive when energy is produced (as for a heater) and negative when energy is consumed (as for a cooler). For `WnsInt`, the losses that heat up the system are positive and the gains that cool down the system are negative.

For stationary models with convection by an incompressible flow, the heat balance becomes:

```
ntfluxInt = QInt
```

which corresponds to the conservation of convective and conductive flux as in:

$$\int_{\partial\Omega_{\text{ext}}}\rho\mathbf{u}E\cdot\mathbf{n}d\sigma - \int_{\partial\Omega_{\text{ext}}}k\nabla T\cdot\mathbf{n}d\sigma = Q_{\text{Int}}$$

ENERGY BALANCE

When the temperature T is solved together with additional mass and momentum equations from Fluid Dynamics for $\mathbf{u}$ and p, the total energy flux also becomes a conserved quantity and the following equation holds (chapter 11.1 in Ref. 8):

$$\frac{d}{dt}\int_{\Omega}\rho E_0 d\omega + \int_{\partial\Omega_{\text{ext}}}(\rho\mathbf{u}H_0 - k\nabla T - \tau\mathbf{u} + \mathbf{q}_{\text{r}})\cdot\mathbf{n}d\sigma = Q_{\text{Int}} + W_{\text{Int}} \tag{14-12}$$

The different variables in this formula are defined in Total Heat Flux and Total Energy Flux. The work sources W_{Int} are the contributions from custom volume forces. Three new variables are then useful to describe the energy powers involved in the system energy balance:

- The *total accumulated energy power* (SI unit: W), `dEi0Int`,
- The *total net energy power* (SI unit: W), `ntefluxInt`, that in this case includes nonconservative work previously in `WnsInt`,
- The *total work source* (SI unit: W), `WInt`.

Their definitions are given in Table 14-2.

TABLE 14-2: GLOBAL POST-PROCESSING VARIABLE FOR TOTAL ENERGY BALANCE

VARIABLE NAME	MATHEMATICAL DEFINITION
`dEi0Int`	$\frac{d}{dt}\int_{\Omega} \rho E_0 d\omega$
`ntefluxInt`	$\int_{\partial\Omega_{\mathrm{ext}}} (\rho \mathbf{u} H_0 - k\nabla T - \tau\mathbf{u} + \mathbf{q}_{\mathrm{r}}) \cdot \mathbf{n} d\sigma$
`QInt`	$\int_{\Omega} Q d\omega + \int_{\partial\Omega_{\mathrm{int}}} Q_{\mathrm{b}} d\omega + \int_{\partial\Omega_{\text{""}}} Q_{\mathrm{r}} d\omega$
`WInt`	$\int_{\Omega} W d\omega$

According to Equation 14-12, the following equality between COMSOL Multiphysics predefined variables holds:

```
dEi0Int + ntefluxInt = QInt + WInt
```

In stationary models, `dEi0Int` is zero so the energy balance simplifies into:

```
ntefluxInt = QInt + WInt
```

At steady-state, and without any additional heat source or volume force (`QInt` and `WInt` equal to zero), the integral of the net energy flux on all boundaries of the flow domain, `ntefluxInt`, vanishes. The corresponding integral of the net heat flux, on the other hand, does not, in general, vanish. It corresponds instead to the losses from mass and momentum equations, such as `WnsInt` for pressure work and viscous heating in fluids. Hence, energy is the conserved quantity, not heat.

Heat Transfer Variables

This section lists some predefined variables that are available for evaluating heat fluxes, sources and integral quantities used in energy balance. All the variable names start with the physics interface prefix. By default the **Heat Transfer** interface prefix is `ht`. As an example, the variable named `tflux` can be analyzed using `ht.tflux` (as long as the physics interface prefix is `ht`).

TABLE 14-3: HEAT FLUX VARIABLES

VARIABLE	NAME	GEOMETRIC ENTITY LEVEL
`dEiInt`	Total Accumulated Heat Power	Global
`ntfluxInt`	Total Net Heat Power	Global
`QInt`	Total Heat Source	Global
`WnsInt`	Total Fluid Losses	Global
`dEi0Int`	Total Accumulated Energy Power	Global
`ntefluxInt`	Total Net Energy Power	Global
`WInt`	Total Work Source	Global

TABLE 14-3: HEAT FLUX VARIABLES

VARIABLE	NAME	GEOMETRIC ENTITY LEVEL
tflux	Total Heat Flux	Domains, boundaries
dflux	Conductive Heat Flux	Domains, boundaries
trlflux	Translational Heat Flux	Domains, boundaries
teflux	Total Energy Flux	Domains, boundaries
ntflux	Normal Total Heat Flux	Boundaries
ndflux	Normal Conductive Heat Flux	Boundaries
ncflux	Normal Convective Heat Flux	Boundaries
ntrlflux	Normal Translational Heat Flux	Boundaries
nteflux	Normal Total Energy Flux	Boundaries
ndflux_u	Internal Normal Conductive Heat Flux, Upside	Interior boundaries
ndflux_d	Internal Normal Conductive Heat Flux, Downside	Interior boundaries
ncflux_u	Internal Normal Convective Heat Flux, Upside	Interior boundaries
ncflux_d	Internal Normal Convective Heat Flux, Downside	Interior boundaries
ntrlflux_u	Internal Normal Translational Heat Flux, Upside	Interior boundaries
ntrlflux_d	Internal Normal Translational Heat Flux, Downside	Interior boundaries
ntflux_u	Internal Normal Conductive Heat Flux, Upside	Interior boundaries
ntflux_d	Internal Normal Conductive Heat Flux, Downside	Interior boundaries
nteflux_u	Internal Normal Total Energy Flux, Upside	Interior boundaries
nteflux_d	Internal Normal Total Energy Flux, Downside	Interior boundaries
Qtot	Domain Heat Sources	Domains
Qbtot	Boundary Heat Sources	Boundaries
Qltot	Line heat source (Line and Point Heat Sources)	Edges, Points (2D, 2Daxi)
Qptot	Point heat source (Line and Point Heat Sources)	Points

GLOBAL VARIABLES

In this paragraph, the variables presented are defined by integrals. A concise notation denotes the different domains of integration: Ω is the geometry domain, $\partial\Omega_{\text{ext}}$ stands for the exterior boundaries and $\partial\Omega_{\text{int}}$ for the interior boundaries.

Total Accumulated Heat Power

The total accumulated heat power variable, dEiInt, is the variation of internal energy per unit time in the domain:

$$\text{dEiInt} = \frac{d}{dt}\int_{\Omega} \rho E d\omega$$

Total Net Heat Power

The total net heat power, `ntfluxInt`, is the integral of Total Heat Flux over all external boundaries:

$$\mathrm{ntfluxInt} = \int_{\partial\Omega_{\mathrm{ext}}} (\rho \mathbf{u} E - k\nabla T + \mathbf{q}_{\mathrm{r}}) \cdot \mathbf{n} d\sigma$$

It thus indicates the sum of incoming and outgoing total heat flux through the system.

Total Heat Source

The total heat source, `QInt`, accounts for all domain sources, interior boundary, edge and point sources, and radiative sources at interior boundaries:

$$\mathrm{QInt} = \int_{\Omega} Q d\omega + \int_{\partial\Omega_{\mathrm{int}}} Q_{\mathrm{b}} d\omega + \int_{\partial\Omega_{\text{""}}} Q_{\mathrm{r}} d\omega$$

Total Fluid Losses

The total fluid losses, `WnsInt`, correspond to the work lost by a fluid by degradation of energy. These works are transmitted to the system through pressure work and viscous heating:

$$\mathrm{WnsInt} = \int_{\Omega} (p_{\mathrm{A}} \nabla \cdot \mathbf{u}) d\omega + \int_{\Omega} (-\tau : \nabla \mathbf{u}) d\omega$$

Total Accumulated Energy Power

The total accumulated energy power, `dEi0Int`, is the variation of total internal energy per unit time in the domain:

$$\mathrm{dEi0Int} = \frac{d}{dt} \int_{\Omega} \rho E_0 d\omega$$

where the total internal energy, E_0, is defined as

$$E_0 = E + \frac{\mathbf{u} \cdot \mathbf{u}}{2}$$

Total Net Energy Power

The total net heat power, `ntefluxInt`, is the integral of Total Energy Flux over all external boundaries:

$$\mathrm{ntefluxInt} = \int_{\partial\Omega_{\mathrm{ext}}} (\rho \mathbf{u} H_0 - k\nabla T - \tau \mathbf{u} + \mathbf{q}_{\mathrm{r}}) \cdot \mathbf{n} d\sigma$$

It thus indicates the sum of incoming and outgoing total energy flux through the system.

Total Work Source

The total work source, `WInt`, sums all work contributions from custom forces:

$$\mathrm{WInt} = \int_{\Omega} W d\omega$$

DOMAIN HEAT FLUXES

On domains the heat fluxes are vector quantities. Their definition can vary depending on the active physics nodes and selected properties.

Total Heat Flux

On domains the total heat flux, `tflux`, corresponds to the conductive and convective heat flux.

For solid domains, for example heat transfer in solids and biological tissue domains, the total heat flux is defined as:

$$\mathrm{tflux} = \mathrm{trlflux} + \mathrm{dflux}$$

For fluid domains (for example, **Heat Transfer in Fluids**), the total heat flux is defined as:

$$\mathrm{tflux} = \mathrm{cflux} + \mathrm{dflux}$$

Conductive Heat Flux

The conductive heat flux variable, `dflux`, is evaluated using the temperature gradient and the effective thermal conductivity:

$$\mathrm{dflux} = -k_{\mathrm{eff}}\nabla T$$

In the general case k_{eff} is the thermal conductivity, k.

For heat transfer in porous media, $k_{\mathrm{eff}} = k_{\mathrm{eq}}$, where k_{eq} is the equivalent conductivity defined in the **Heat Transfer in Porous Media** feature.

The **Heat Transfer in Porous Media** feature requires one of the following products: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Heat Transfer Module, or Subsurface Flow Module.

Translational Heat Flux

Similar to convective heat flux but defined for solid domains with translation. The variable name is `trlflux`.

Total Energy Flux

The total energy flux, `teflux`, is defined when viscous heating is enabled:

$$\mathrm{teflux} = \rho\mathbf{u}H_0 + \mathrm{dflux} + \tau\cdot\mathbf{u}$$

where the total enthalpy, H_0, is defined as

$$H_0 = H + \frac{\mathbf{u}\cdot\mathbf{u}}{2}$$

BOUNDARY HEAT FLUXES

All the domain heat fluxes (vector quantity) are also available as boundary heat fluxes. The boundary heat fluxes are then equal to the mean value of the heat fluxes on adjacent domains. In addition, normal boundary heat fluxes (scalar quantity) are available on boundaries.

Normal Total Heat Flux

The variable `ntflux` is defined as:

$$\mathrm{ntflux} = \mathrm{ndflux} + \mathrm{ncflux} + \mathrm{ntrlflux}$$

Normal Conductive Heat Flux

The variable `ndflux` is defined by on exterior boundaries as:

- ndflux = –dflux_spatial(T) if the adjacent domain is on the downside,
- ndflux = –uflux_spatial(T) if the adjacent domain is on the upside,

and, on interior boundaries, as:
ndflux = (uflux_spatial(T) – dflux_spatial(T))/2

Normal Convective Heat Flux

The variable `ncflux` is defined as:

$$\mathrm{ncflux} = \mathrm{mean}(\mathrm{cflux})\cdot\mathbf{n}$$

Normal Translational Heat Flux

The variable `ntrlflux` is defined as

$$\mathrm{ntrlflux} = \mathrm{mean}(\mathrm{trlflux})\cdot\mathbf{n}$$

Normal Total Energy Flux

The variable `nteflux` is defined as:

$$\mathrm{nteflux} = \mathrm{mean}(\mathrm{teflux})\cdot\mathbf{n} - \mathrm{mean}(\mathrm{dflux})\cdot\mathbf{n} + \mathrm{ndflux}$$

INTERNAL BOUNDARY HEAT FLUXES

The internal normal boundary heat fluxes (scalar quantity) are available on interior boundaries. They are calculated using the upside and the downside value of heat fluxes from the adjacent domains.

Internal Normal Conductive Heat Flux, Upside

The variable `ndflux_u` is defined as:

$$\mathrm{ndflux\_u} = \mathrm{uflux\_spatial}(T)$$

Internal Normal Conductive Heat Flux, Downside

The variable `ndflux_d` is defined as:

$$\mathrm{ndflux\_d} = \mathrm{dflux\_spatial}(T)$$

Internal Normal Convective Heat Flux, Upside

The variable `ncflux_u` is defined as:

$$\mathrm{ncflux\_u} = \mathrm{up}(\mathrm{cflux})\cdot\mathbf{un}$$

Internal Normal Convective Heat Flux, Downside

The variable `ncflux_d` is defined as:

$$\mathrm{ncflux\_d} = \mathrm{down}(\mathrm{cflux})\cdot\mathbf{dn}$$

Internal Normal Translational Heat Flux, Upside

The variable `ntrlflux_u` is defined as:

$$\mathrm{ntrlflux\_u} = \mathrm{up}(\mathrm{trlflux})\cdot\mathbf{un}$$

Internal Normal Translational Heat Flux, Downside

The variable `ntrlflux_d` is defined as:

$$\mathrm{ntrlflux\_d} = \mathrm{down}(\mathrm{trlflux})\cdot\mathbf{dn}$$

Internal Normal Total Heat Flux, Upside

The variable `ntflux_u` is defined as:

$$\mathrm{ntflux\_u} = \mathrm{ndflux\_u} + \mathrm{ncflux\_u} + \mathrm{ntrlflux\_u}$$

Internal Normal Total Heat Flux, Downside

The variable `ntflux_d` is defined as:

$$\mathrm{ntflux\_d} = \mathrm{ndflux\_d} + \mathrm{ncflux\_d} + \mathrm{ntrlflux\_d}$$

Internal Normal Total Energy Flux, Upside

The variable `nteflux_u` is defined as:

$$\text{nteflux\_u} = \text{up(teflux)} \cdot \mathbf{un} - \text{up(dflux)} \cdot \mathbf{un} + \text{ndflux\_u}$$

Internal Normal Total Energy Flux, Downside

The variable `nteflux_d` is defined as:

$$\text{nteflux\_d} = \text{down(teflux)} \cdot \mathbf{dn} - \text{down(dflux)} \cdot \mathbf{dn} + \text{ndflux\_d}$$

DOMAIN HEAT SOURCES

The sum of the domain heat sources added by different physics features is available in the variable, Q_{tot} (SI unit: $\mathrm{W/m^3}$). This variable `Qtot` is the sum of:

- Q's which are the heat sources added by the Heat Source(described for the **Heat Transfer** interface) and Electromagnetic Heat Source (described for the Joule Heating interface) features.

BOUNDARY HEAT SOURCES

The sum of the boundary heat sources added by different boundary conditions is available in the variable, $Q_{\mathrm{b,tot}}$ (SI unit: $\mathrm{W/m^2}$). This variable `Qbtot` is the sum of:

- Q_{b} which is the boundary heat source added by the Boundary Heat Source boundary condition.
- Q_{sh} which is the boundary heat source added by the Boundary Electromagnetic Heat Source boundary condition (described for the Joule Heating interface).

LINE AND POINT HEAT SOURCES

The sum of the line heat sources is available in a variable called `Qltot` (SI unit: W/m).

The sum of the point heat sources is available in a variable called `Qptot` (SI unit: W).

About the Boundary Conditions for the Heat Transfer Interfaces

TEMPERATURE AND HEAT FLUX BOUNDARY CONDITIONS

The heat equation accepts two basic types of boundary conditions: specified temperature and specified heat flux. The specified condition is of constraint type and prescribes the temperature on a boundary:

$$T = T_0 \qquad \text{on } \partial\Omega$$

while the latter specifies the inward heat flux

$$-\mathbf{n} \cdot \mathbf{q} = q_0 \qquad \text{on } \partial\Omega$$

where

- $\mathbf{q}$ is the *conductive heat flux vector* (SI unit: $\mathrm{W/m^2}$), $\mathbf{q} = -k\nabla T$.
- $\mathbf{n}$ is the normal vector on the boundary.
- q_0 is the *inward heat flux* (SI unit: $\mathrm{W/m^2}$), normal to the boundary.

The inward heat flux, q_0, is often a sum of contributions from different heat transfer processes (for example, radiation and convection). The special case $q_0 = 0$ is called *thermal insulation*.

A common type of heat flux boundary conditions is one for which $q_0 = h \cdot (T_{\mathrm{inf}} - T)$, where T_{inf} is the temperature far away from the modeled domain and the heat transfer coefficient, h, represents all the physics occurring between the boundary and “far away.” It can include almost anything, but the most common situation is that h represents the effect of an exterior fluid cooling or heating the surface of a solid, a phenomenon often referred to as convective cooling or heating.

Convective heat flux requires either the Heat Transfer Module or the CFD Module.

OVERRIDING MECHANISM FOR HEAT TRANSFER BOUNDARY CONDITIONS

This section includes information for features that might require additional modules.

Many boundary conditions are available in heat transfer. Some of them can coexist (for example, Heat Flux and Highly Conductive Layer). Others cannot coexist (for example, Heat Flux and Thermal Insulation).

Several categories of boundary condition exist in heat transfer. Table 14-4 gives the overriding rules for these groups.

1. Temperature, Convective Outflow, Open Boundary, Inflow Heat Flux
2. Thermal Insulation, Symmetry, Periodic Heat Condition
3. Highly Conductive Layer
4. Heat Flux, Convective Heat Flux
5. Boundary Heat Source, Radiation Group
6. Surface-to-Surface Radiation, Diffuse Mirror, Prescribed Radiosity, Surface-to-Ambient Radiation
7. Opaque Surface, Incident Intensity, Continuity on Interior Boundaries
8. Thin Thermally Resistive Layers, Thermal Contact

TABLE 14-4: OVERRIDING RULES FOR HEAT TRANSFER BOUNDARY CONDITIONS

A\B	1	2	3	4	5	6	7	8
1-Temperature	X	X	X					X
2-Thermal Insulation	X	X				X		
3-Highly Conductive Layer	X		X					
4-Heat Flux	X	X						
5-Boundary heat source								
6-Surface-to-surface radiation		X				X		
7-Opaque Surface							X	
8-Thin Thermally Resistive Layer	X							X

When there is a boundary condition *A* above a boundary condition *B* in the model tree and both conditions apply to the same boundary, use Table 14-4 to determine if *A* is overridden by *B* or not:

- Locate the line that corresponds to the *A* group (see above the definition of the groups). In the table above only the first member of the group is displayed.

- Locate the column that corresponds to the group of *B*.
- If the corresponding cell is empty *A* and *B* contribute. If it contains an X, *B* overrides A.

Group 4 and group 5 boundary conditions are always contributing. That means that they never override any other boundary condition. But they might be overridden.

Example 1

Surface-to-Surface radiation requires the Heat Transfer Module.

Consider a boundary where **Temperature** is applied. Then a **Surface-to-Surface Radiation** boundary condition is applied on the same boundary afterward.

- **Temperature** belongs to group 1.
- **Surface-to-surface radiation** belongs to group 6.
- The cell on the line of group 1 and the column of group 6 is empty so **Temperature** and **Surface-to-Surface radiation** contribute.

Example 2

Convective Heat Flux requires either the Heat Transfer Module or the CFD Module.

Consider a boundary where **Convective Heat Flux** is applied. Then a **Symmetry** boundary condition is applied on the same boundary afterward.

- **Convective Heat Flux** belongs to group 4.
- **Symmetry** belongs to group 2.
- The cell on the line of group 4 and the column of group 2 contains an X so **Convective Heat Flux** is overridden by **Symmetry**.

In Example 2 above, if **Symmetry** followed by **Convective Heat Flux** is added, the boundary conditions contribute.

Radiative Heat Transfer in Transparent Media

This discussion so far has considered heat transfer by means of conduction and convection. A third mechanism for heat transfer is radiation. Consider an environment with fully transparent or fully opaque objects. Thermal radiation denotes the stream of electromagnetic waves emitted from a body at a certain temperature.

DERIVING THE RADIATIVE HEAT FLUX

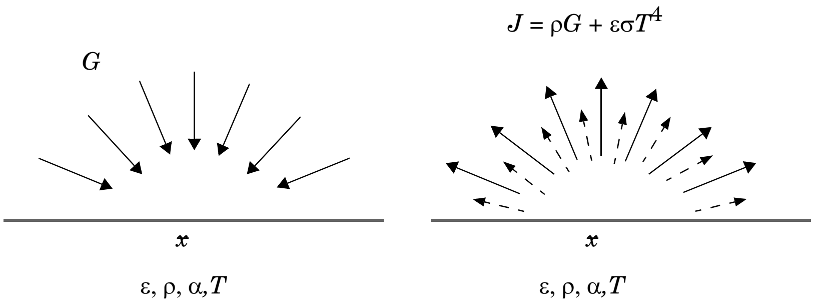


Figure 14-1: Arriving irradiation (left), leaving radiosity (right).

Consider Figure 14-1. A point $\bar{x}$ is located on a surface that has an emissivity ε, reflectivity ρ, absorptivity α, and temperature T. Assume that the body is opaque, which means that no radiation is transmitted through the body. This is true for most solid bodies.

The total arriving radiative flux at $\bar{x}$ is named the irradiation, G. The total outgoing radiative flux $\bar{x}$ is named the radiosity, J. The radiosity is the sum of the reflected radiation and the emitted radiation:

$$J = \rho G + \varepsilon\sigma T^4 \qquad (14\text{-}13)$$

The net inward radiative heat flux, q, is then given by the difference between the irradiation and the radiosity:

$$q = G - J \qquad (14\text{-}14)$$

Using Equation 14-13 and Equation 14-14 J can be eliminated and a general expression is obtained for the net inward heat flux into the opaque body based on G and T.

$$q = (1-\rho)G - \varepsilon\sigma T^4 \qquad (14\text{-}15)$$

Most opaque bodies also behave as ideal gray bodies, meaning that the absorptivity and emissivity are equal, and the reflectivity is therefore obtained from the following relation:

$$\alpha = \varepsilon = 1-\rho \qquad (14\text{-}16)$$

Thus, for ideal gray bodies, q is given by:

$$q = \varepsilon(G - \sigma T^4) \qquad (14\text{-}17)$$

This is the expression used for the radiation boundary condition.

RADIATION TYPES

It is common to differentiate between two types of radiative heat transfer: surface-to-ambient radiation and *surface-to-surface radiation*. Equation 14-17 holds for both radiation types, but the irradiation term, G, is different for each of them. The **Heat Transfer** interface supports radiation.

Surface-to-surface radiation requires the Heat Transfer Module.

SURFACE-TO-AMBIENT RADIATION

Surface-to-ambient radiation assumes the following:

- The ambient surroundings in view of the surface have a constant temperature, T_{amb}.
- The ambient surroundings behave as a blackbody. This means that the emissivity and absorptivity are equal to 1, and the reflectivity is 0.

These assumptions allow the irradiation to be explicitly expressed as

$$G = \sigma T_{\mathrm{amb}}^4 \qquad (14\text{-}18)$$

Inserting Equation 14-18 into Equation 14-17 results in the net inward heat flux for surface-to-ambient radiation

$$q = \varepsilon\sigma(T_{\mathrm{amb}}^4 - T^4) \qquad (14\text{-}19)$$

For boundaries where a surface-to-ambient radiation is specified, COMSOL Multiphysics adds this term to the right-hand side of Equation 14-19.

Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces

The different versions of the **Heat Transfer** interface have the advanced option to set the stabilization method parameters. This section provides information pertaining to these options. To display the stabilization sections, click the **Show** button (👁) and select **Stabilization**.

CONSISTENT STABILIZATION

This section contains two consistent stabilization methods: streamline diffusion and crosswind diffusion. These are consistent stabilization methods, which means that they do not perturb the original transport equation.

The consistent stabilization methods are activate by default. A stabilization method is active when the corresponding check box is selected.

Streamline Diffusion

Streamline diffusion is active by default and should remain active for optimal performance for heat transfer in fluids or other applications that include a convective or translational term.

Crosswind Diffusion

Streamline diffusion introduces artificial diffusion in the streamline direction. This is often enough to obtain a smooth numerical solution provided that the exact solution of the heat equation does not contain any discontinuities. At sharp gradients, however, undershoots and overshoots can occur in the numerical solution. Crosswind diffusion addresses these spurious oscillations by adding diffusion orthogonal to the streamline direction—that is, in the crosswind direction.

INCONSISTENT STABILIZATION

This section contains a single stabilization method: isotropic diffusion. Adding isotropic diffusion is equivalent to adding a term to the physical diffusion coefficient. This means that the original problem is not solved, which is why isotropic diffusion is an inconsistent stabilization method. Although, the added diffusion definitely attenuates spurious oscillations, try to minimize the use of isotropic diffusion.

By default there is no isotropic diffusion. To add isotropic diffusion, select the **Isotropic diffusion** check box. The field for the tuning parameter δ_{id} then becomes available. The default value is 0.25; increase or decrease the value of δ_{id} to increase or decrease the amount of isotropic diffusion.

- Stabilization Techniques
- Stabilization

References for the Heat Transfer Interfaces

1. F.P. Incropera, D.P. DeWitt, T.L. Bergman and A.S. Lavine, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, 6th ed., 2006.

2. R. Codina, “Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation,” *Comp. Meth.Appl. Mech. Engrg*, vol. 156, pp. 185–210, 1998.

3. A. Bejan, *Heat Transfer*, John Wiley & Sons, 1993.

4. G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 2000.

5. R.L. Panton, *Incompressible Flow*, 2nd ed., John Wiley & Sons, 1996.

6. M. Kaviany, *Principles of Convective Heat Transfer*, 2nd ed., Springer, 2001.

7. T. Poinsot and D. Veynante, *Theoretical and Numerical Combustion*, 2nd ed., Edwards, 2005.

8. R.B. Bird, W.E. Stewart and E.N. Lightfoot, Transport Phenomena, 2nd ed., John Wiley & Sons, 2007.

9. W. Wagner, and H-J Kretzschmar, *International Steam Tables*, 2nd ed., Springer, 2008.

The Heat Transfer Interface

After selecting a version of the physics interface , default nodes are added under the main node, which then defines which version of the Heat Transfer interface is added.

Heat Transfer in Solids

The **Heat Transfer in Solids** ( ) interface is used to model heat transfer by conduction, convection, and radiation. A **Heat Transfer in Solids** model is active by default on all domains. All functionality for including other domain types, such as a fluid domain, is also available.

The temperature equation defined in solid domains corresponds to the differential form of the Fourier's law that may contain additional contributions like heat sources.

When this version of the physics interface is added, these default nodes are added to the **Model Builder**—**Heat Transfer in Solids**, **Thermal Insulation** (the default boundary condition), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and sources. You can also right-click **Heat Transfer is Solids** to select physics from the context menu.

Heat Transfer in Fluids

The **Heat Transfer in Fluids** ( ) is used to model heat transfer by conduction, convection, and radiation. A **Heat Transfer in Fluids** model is active by default on all domains. All functionality for including other domain types, such as a solid domain, is also available.

The temperature equation defined in fluid domains corresponds to the convection-diffusion equation that may contain additional contributions like heat sources.

When this version of the physics interface is added, these default nodes are added to the **Model Builder**—**Heat Transfer in Fluids**, **Thermal Insulation** (the default boundary condition), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and sources. You can also right-click **Heat Transfer in Fluids** to select physics from the context menu.

Benefits of the Different Heat Transfer Interfaces

The benefit of the different versions of the **Heat Transfer** interfaces, with `ht` as the common default identifier, is that it is easy to add the default settings when selecting the interface. At any time, add a **Heat Transfer in Fluids** or **Heat Transfer in Solids** node from the **Physics** toolbar—the functionality is always available.

The **Joule Heating** ( ) multiphysics interface is also available. See The Joule Heating Interface for information.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to these variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first physics interface in the model) is `ht`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define heat transfer and a temperature field. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

This section does not require any input.

CONSISTENT STABILIZATION

To display this section, click the **Show** button ( ) and select **Stabilization**. The **Streamline diffusion** check box is selected by default and should remain selected for optimal performance for heat transfer in fluids or other applications that include a convective or translational term. **Crosswind diffusion** provides extra diffusion in regions with sharp gradients. The added diffusion is orthogonal to the streamlines, so streamline diffusion and crosswind diffusion can be used simultaneously. The **Crosswind diffusion** check box is selected by default.

INCONSISTENT STABILIZATION

To display this section, click the **Show** button ( ) and select **Stabilization**. The **Isotropic diffusion** check box is not selected by default.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**.

- Select an element order (shape function order) for the **Temperature**—**Quadratic** (the default), **Linear**, **Cubic**, **Quartic**, or **Quintic**.
- The **Compute boundary fluxes** check box is selected by default so that COMSOL Multiphysics computes accurate boundary flux variables.
- The **Apply smoothing to boundary fluxes** check box is selected by default. The smoothing can provide a more well-behaved flux value close to singularities.
- In the table, specify the **Value type when using splitting of complex variables**—**Real** (the default) or **Complex**.

DEPENDENT VARIABLES

The **Heat Transfer** interfaces have the dependent variable **Temperature** T.The dependent variable names can be changed. Editing the name of a scalar dependent variable changes both its field name and the dependent variable name. If a new field name coincides with the name of another field of the same type, the fields share degrees of freedom and dependent variable names. A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

ADVANCED SETTINGS

Add both a **Heat Transfer (ht)** and a **Moving Mesh (ale)** interface (found under the **Mathematics>Deformed Mesh** branch when adding a physics interface) then click the **Show** button ( ) and select **Advanced Physics Options** to display this section.

When the component contains a moving mesh, the **Enable conversions between material and spatial frame** check box is selected by default.

This option has no effect when the component does not contain a moving frame since the material and spatial frames are identical in such cases. With a moving mesh, and when this option is active, the heat transfer features automatically account for deformation effects on heat transfer properties. In particular the effects of volume changes on the density are considered. Rotation effects on the thermal conductivity of an anisotropic material and, more generally, deformation effects on an arbitrary thermal conductivity, are also covered. When the **Enable conversions between material and spatial frame** check box is not selected, the feature inputs (for example, Heat

Source, Heat Flux, Boundary Heat Source, and Line Heat Source) are not converted and are instead defined on the **Spatial** frame.

- About Frames in Heat Transfer
- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer Interfaces
- Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces
- Heat Transfer Theory
- Stabilization

Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer Interfaces

The Heat Transfer Interface has these domain, boundary, edge, point, and pair nodes and subnodes () available. These nodes, listed in alphabetical order, are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. However, to add subnodes, right-click the parent node.

- Boundary Heat Source
- Continuity
- Heat Flux
- Heat Source
- Heat Transfer in Fluids
- Heat Transfer in Solids
- Initial Values
- Line Heat Source
- Line Heat Source on Axis
- Outflow
- Periodic Heat Condition
- Point Heat Source
- Point Heat Source on Axis
- Surface-to-Ambient Radiation
- Symmetry
- Temperature
- Thermal Insulation (the default boundary condition)
- Thin Thermally Resistive Layer
- Translational Motion

If you also have the Heat Transfer Module, there are several other feature nodes available and described in the *Heat Transfer Module User's Guide*.

For axisymmetric components, COMSOL Multiphysics takes the axial symmetry boundaries into account and automatically adds an **Axial Symmetry** node that is valid on the axial symmetry boundaries only.

Heat Transfer in Solids

The **Heat Transfer in Solids** node uses the heat equation, Equation 14-20, to model heat transfer in solids:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \tag{14-20}$$

For a steady-state problem the temperature does not change with time and the first term disappears. The equation includes the material properties: density ρ, heat capacity C_p, and thermal conductivity k (a scalar or a tensor when the thermal conductivity is anisotropic), and a heat source (or sink) Q—one or more heat sources can be added separately.

When parts of the model are moving in the material frame, right-click the **Heat Transfer in Solids** node to add a Translational Motion subnode to take this into account.

With the CFD Module, right-click to add a **Pressure Work** subnode.

With the Heat Transfer Module, right-click to add a **Pressure Work** or an **Opaque** subnode. The **Opaque** subnode is automatically added to the entire selection when **Surface-to-surface radiation** is activated. The selection can be edited.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and can not be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

MODEL INPUTS

This section contains fields and values that are inputs to expressions defining material properties. If such user-defined materials are added, the model inputs appear here. Initially, this section is empty.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the component includes (except for boundary coordinate systems). The coordinate system is used to define directions for orthotropic and anisotropic thermal conductivities.

HEAT CONDUCTION, SOLID

The default setting is to use the **Thermal conductivity** k (SI unit: W/(m·K)) **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the thermal conductivity, and enter another value or expression.

The thermal conductivity describes the relationship between the heat flux vector $\mathbf{q}$ and the temperature gradient ∇T as in $\mathbf{q} = -k\nabla T$, which is Fourier’s law of heat conduction. Enter this quantity as power per length and temperature.

The components of the thermal conductivity k when given on tensor form (k_{xx}, k_{yy}, and so on, representing an anisotropic thermal conductivity) are available as `ht.kxx`, `ht.kyy`, and so on (using the default interface identifier `ht`). The single scalar mean effective thermal conductivity `ht.kmean` is the mean value of the diagonal elements k_{xx}, k_{yy}, and k_{zz}.

Fourier's law assumes that the thermal conductivity tensor is symmetric. A non symmetric tensor can lead to unphysical results.

THERMODYNAMICS, SOLID

The default **Density** ρ (SI unit: kg/m^3) and **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)) use values **From material**. Select **User defined** to enter other values or expressions. The heat capacity at constant pressure describes the amount of heat energy required to produce a unit temperature change in a unit mass.

Thermal Diffusivity

In addition, the thermal diffusivity α, defined as $k/(\rho C_p)$ (SI unit: m^2/s), is also a predefined quantity. The thermal diffusivity can be interpreted as a measure of thermal inertia (heat propagates slowly where the thermal diffusivity is low, for example). The components of the thermal diffusivity α, when given on tensor form (α_{xx}, α_{yy}, and so on, representing an anisotropic thermal diffusivity) are available as `ht.alphaTdxx`, `ht.alphaTdyy`, and so on (using the default interface identifier `ht`). The single scalar mean thermal diffusivity `ht.alphaTdMean` is the mean value of the diagonal elements α_{xx}, α_{yy}, and α_{zz}. The denominator ρC_p is the effective volumetric heat capacity which is also available as a predefined quantity, `ht.C_eff`.

- Axisymmetric Transient Heat Transfer: model library path **COMSOL_Multiphysics/Heat_Transfer/heat_transient_axi**
- 2D Heat Transfer Benchmark with Convective Cooling: model library path **COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d**

Translational Motion

Right-click the Heat Transfer in Solids node to add the **Translational Motion** subnode, which provides movement by translation to the model for heat transfer in solids. It adds the following contribution to the right-hand side of Equation 14-20, defined in the parent node:

$$-\rho C_{\mathrm{p}}\mathbf{u}\cdot\nabla T$$

The contribution describes the effect of a moving coordinate system which is required to model, for example, a moving heat source.

Special care must be taken on boundaries where $\mathbf{n}\cdot\mathbf{u}\neq 0$. The Heat Flux boundary condition does not, for example, work on boundaries where $\mathbf{n}\cdot\mathbf{u} < 0$.

DOMAIN SELECTION

From the **Selection** list, choose the domains on which to apply the translational motion.

By default, the selection is the same as for the Heat Transfer in Solids node that it is attached to, but it is possible to use more than one Heat Translation subnode, each covering a subset of the Heat Transfer in Solids node's selection.

TRANSLATIONAL MOTION

Enter values for x, y, and z (in 3D) components of the **Velocity field** $\mathbf{u}_{\mathrm{trans}}$ (SI unit: m/s).

Heat Transfer in Fluids

The **Heat Transfer in Fluids** interface uses the following version of the heat equation to model heat transfer in fluids:

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q \qquad (14\text{-}21)$$

For a steady-state problem the temperature does not change with time and the first term disappears. This equation includes the following material properties, fields, and sources:

- **Density** ρ (SI unit: kg/m^3)
- **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K))—describes the amount of heat energy required to produce a unit temperature change in a unit mass.
- **Thermal conductivity** k (SI unit: W/(m·K))—a scalar or a tensor if the thermal conductivity is anisotropic.
- **Velocity field** $\mathbf{u}$ (SI unit: m/s)—either an analytic expression or a velocity field from a fluid-flow interface.
- The heat source (or sink) Q—one or more heat sources can be added separately.
- The **Ratio of specific heats** γ (dimensionless)— the ratio of the heat capacity at constant pressure, C_p, to heat the capacity at constant volume, C_v.

When using the ideal gas law to describe a fluid, specifying γ is sufficient to evaluate C_p. For common diatomic gases such as air, $\gamma = 1.4$ is the standard value. Most liquids have $\gamma = 1.1$ while water has $\gamma = 1.0$. γ is used in the streamline stabilization and in the variables for heat fluxes and total energy fluxes.

Heat Transfer by Free Convection: model library path **COMSOL_Multiphysics/Multiphysics/free_convection**

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and can not be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

MODEL INPUTS

This section has fields and values that are inputs to expressions that define material properties. If such user-defined property groups are added, the model inputs appear here.

There are also two standard model inputs—**Absolute pressure** and **Concentration**. The absolute pressure is used in some predefined quantities that include the enthalpy (the energy flux, for example).

Absolute Pressure

This section controls the variable itself as well as any property value (reference pressures) used when solving for the pressure. There are usually two ways to calculate the pressure when describing fluid flow with mass- and heat transfer. Solve for the absolute pressure or a pressure (often denoted gauge pressure) that relates to the absolute pressure through a reference pressure.

Which option to choose usually depends on the system and the equations being solved. For example, in a unidirectional incompressible flow problem, the pressure drop over the modeled domain is probably many orders of magnitude smaller than the atmospheric pressure, which, when included, reduces the stability and convergence properties of the solver. In other cases, such as when pressure is part of an expression for the gas volume or the diffusion coefficients, you might need to solve for the absolute pressure.

The default **Absolute pressure** p_A (SI unit: Pa) is **User defined** and is 1 atm (101,325 Pa). When additional physics interfaces are added to the model, the pressure variables solved for can also be selected from the list. For example, if a fluid-flow interface is added you can select **Pressure (spf/fp)** from the list.

When a **Pressure** variable is selected, the **Reference pressure** check box is selected by default and the default value of p_{ref} is `1[atm]` (1 atmosphere).

This makes it possible to use a system-based (gauge) pressure as the pressure variable while automatically including the reference pressure in places where it is required, such as for gas flow governed by the gas law. While this check box maintains control over the pressure variable and instances when the absolute pressure is required within this specific physics interface, it might not do so with other physics interfaces that it is coupled to. In such models, check the coupling between any physics interfaces using the same variable.

Velocity Field

The default **Velocity field u** (SI unit: m/s) is **User defined**. When **User defined** is selected, enter values or expressions for the components based on space dimensions. The defaults are 0 m/s. Or select an existing velocity field in the component (for example, **Velocity field (spf/fp1)** from a Laminar Flow interface).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the component includes (except for boundary coordinate systems). The coordinate system is used to define directions for orthotropic and anisotropic thermal conductivities.

HEAT CONDUCTION, FLUID

The default **Thermal conductivity** k (SI unit: W/(m·K)) is taken **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the thermal conductivity, and enter another value or expression.

The thermal conductivity describes the relationship between the heat flux vector $\mathbf{q}$ and the temperature gradient ∇T as in $\mathbf{q} = -k\nabla T$ which is Fourier’s law of heat conduction. Enter this quantity as power per length and temperature.

THERMODYNAMICS, FLUID

The only available **Fluid type** is **Gas/Liquid**.

The default **Density** ρ (SI unit: kg/m^3), **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)), and **Ratio of specific heats** γ (dimensionless) for a general gas or liquid use values **From material**. Select **User defined** to enter other values or expressions.

Initial Values

The **Initial Values** node adds an initial value for the temperature that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. Add additional **Initial Values** nodes from the **Physics** toolbar.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and can not be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the initial value of the **Temperature** T (SI unit: K). The default value is approximately room temperature, 293.15 K (20 °C).

Heat Source

The **Heat Source** describes heat generation within the domain. You express heating and cooling with positive and negative values, respectively. Add one or more nodes as required—all heat sources within a domain contribute to the total heat source. Specify the heat source as the heat per unit volume, as a linear heat source, or as a total heat source (power).

DOMAIN SELECTION

From the **Selection** list, choose the domains to add the heat source to.

HEAT SOURCE

Click the **General source** (the default), **Linear source**, or **Total power** button.

- If **General source** is selected, enter a value for the distributed heat source Q (SI unit: W/m^3) when the default option, **User defined**, is selected. The default is 0 W/m^3(that is, no heat source). See also Additional General Source Options.
- If **Linear source** ($Q = q_s{\cdot}T$) is selected, enter the **Production/absorption coefficient** q_s (SI unit: W/(m^3·K)). The default is 0 W/(m^3·K).
- If **Total power** is selected, enter the total heat source, P_{tot}, (SI unit: W). The default is 0 W. In this case $Q = P_{tot}/V$, where V is the total volume of the selected domains.

In 3D and 2D axial symmetry, $V = \int 1$.

In 2D and 1D axial symmetry:

$$V = dz\int 1$$

where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available for defining dz.

In 1D:

$$V = A_c\int 1$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available for defining A_c.

The advantage of writing the source on the second form is that it can be accounted for in the streamline diffusion stabilization. The stabilization applies when q_s is independent of the temperature, but some stability can be gained as long as q_s is only weakly dependent on the temperature.

Additional General Source Options

For the general heat source Q, there are predefined heat sources available (in addition to a **User defined** heat source) when simulating heat transfer together with electrical or electromagnetic physics interfaces. Such sources represent, for example, ohmic heating and induction heating.

The following options are also available from the **General source** list above but require additional physics interfaces and/or licenses as indicated.

- With the addition of an Electric Currents interface, the **Total power dissipation density (ec/cucn1)** heat source is available from the General source list.
- With the addition of any version of the Electromagnetic Waves interface (which requires the RF Module), the **Total power dissipation density (emw/wee1)** and **Electromagnetic power loss density (emw/wee1)** heat sources are available from the General source list.
- With the addition of a Magnetic Fields interface (a 3D model requires the AC/DC Module), the **Electromagnetic heating (mf/al1)** heat source is available from the General source list.
- With the addition of a Magnetic and Electric Fields interface (which requires the AC/DC Module), the **Electromagnetic heating (mef/alc1)** heat source is available from the General source list.
- For the Heat Transfer in Porous Media interface, with the addition of interfaces from the Batteries & Fuel Cells Module, Corrosion Module, or Electrodeposition Module, heat sources from the electrochemical current distribution interfaces are available.

FRAME SELECTION

To display this section, add both a **Heat Transfer (ht)** and a **Moving Mesh (ale)** interface (found under the **Mathematics>Deformed Mesh** branch when adding a physics interface). Then click the **Show** button ( ) and select **Advanced Physics Options**.

When the model contains a moving mesh, the **Enable conversions between material and spatial frame** check box is selected by default in the Heat Transfer interface, which in turn enables further options. Use **Frame Selection** to select the frame where the input variables are defined. If **Spatial** is selected, the variables take their values from the edit fields. If **Material** (the default) is selected, a conversion from the material to the spatial frame is applied to the edit field values**.**

- About Frames in Heat Transfer
- The Heat Transfer Interface
- Stabilization Techniques

Thermal Insulation

The **Thermal Insulation** node is the default boundary condition for all Heat Transfer interfaces. This boundary condition means that there is no heat flux across the boundary:

$$\mathbf{n}\cdot(k\nabla T) = 0$$

and hence specifies where the domain is well insulated. Intuitively, this equation says that the temperature gradient across the boundary is zero. For this to be true, the temperature on one side of the boundary must equal the temperature on the other side. Because there is no temperature difference across the boundary, heat cannot transfer across it.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and can not be edited; that is, the selection is automatically made and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

Temperature

Use the **Temperature** node to specify the temperature somewhere in the geometry, for example, on boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to define the temperature.

TEMPERATURE

The equation for this condition is $T = T_0$ where T_0 is the prescribed temperature on the boundary. Enter the value or expression for the **Temperature** T_0 (SI unit: K). The default is 293.15 K.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**.

- By default **Classic constraints** is selected. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required.

- Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.
- Select the **Discontinuous Galerkin constraints** button when **Classic constraints** do not work satisfactorily.

The **Discontinuous Galerkin constraints** option is especially useful to prevent oscillations on inlet boundaries where convection dominates. Unlike the **Classic constraints**, these constraints do not enforce the temperature on the boundary extremities. This is relevant on fluid inlets where the temperature condition should not be enforced on the walls at the inlet extremities.

Show More Physics Options

Outflow

The **Outflow** node provides a suitable boundary condition for convection-dominated heat transfer at outlet boundaries. In a model with convective heat transfer, this condition states that the only heat transfer occurring across the boundary is by convection. The temperature gradient in the normal direction is zero, and there is no radiation. This is usually a good approximation of the conditions at an outlet boundary in a heat transfer model with fluid flow.

BOUNDARY SELECTION

In most cases, the **Outflow** node does not require any user input. If required, select the boundaries that are convection-dominated outlet boundaries.

Symmetry

The **Symmetry** node provides a boundary condition for symmetry boundaries. This boundary condition is similar to a Thermal Insulation condition, and it means that there is no heat flux across the boundary.

BOUNDARY SELECTION

In most cases, the node does not require any user input. If required, define the symmetry boundaries.

Heat Flux

Use the **Heat Flux** node to add heat flux across boundaries. A positive heat flux adds heat to the domain. This feature is not applicable to inlet boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

HEAT FLUX

Click to select the **General inward heat flux** (the default), **Inward heat flux**, or **Total heat flux** button.

General Inward Heat Flux

If **General inward heat flux** q_0 (SI unit: W/m^2) is selected, it adds to the total flux across the selected boundaries. Enter a value for q_0 to represent a heat flux that enters the domain. For example, any electric heater is well represented by this condition, and its geometry can be omitted. The default is 0 W/m^2.

Inward Heat Flux

If **Inward heat flux** is selected, enter the **Heat transfer coefficient** h (SI unit: W/(m^2·K)). The default is 0 W/(m^2·K). Also enter an **External temperature** T_{ext} (SI unit: K). The default is 293.15 K. The value depends on the geometry and the ambient flow conditions. Inward heat flux is defined by $q_0 = h(T_{\text{ext}} - T)$.

For a thorough introduction about how to calculate heat transfer coefficients, see Incropera and DeWitt in Ref. 1.

Total Heat Flux

If **Total heat flux** is selected, enter the total heat flux q_{tot} (SI unit: W) across the boundaries where the **Heat Flux** node is active. The default is 0 W. In this case $q_0 = q_{\text{tot}}/A$, where A is the total area of the selected boundaries.

In 3D and 2D axial symmetry, $A = \int 1$ ·

In 2D and 1D axial symmetry:

$$A = dz\int 1$$

where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available to define dz.

In 1D:

$$A = A_c\int 1$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available to define A_c.

FRAME SELECTION

The settings are the same as for the Heat Source node and are described under the corresponding **Frame Selection** section.

- About Frames in Heat Transfer
- The Heat Transfer Interface

Surface-to-Ambient Radiation

Use the **Surface-to-Ambient Radiation** condition to add surface-to-ambient radiation to boundaries. The net inward heat flux stemming from surface-to-ambient radiation is

$$q = \varepsilon\sigma(T_{\mathrm{amb}}^4 - T^4)$$

where ε is the surface emissivity, σ is the Stefan-Boltzmann constant (a predefined physical constant), and T_{amb} is the ambient temperature.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

MODEL INPUTS

This section contains fields and values that are inputs to expressions defining material properties. If such user-defined materials are added, the model inputs appear here. Initially, this section is empty.

SURFACE-TO-AMBIENT RADIATION

The default **Surface emissivity** ε (a dimensionless number between 0 and 1) is taken **From material**. An emissivity of 0 means that the surface emits no radiation at all and an emissivity of 1 means that it is a perfect blackbody.

Enter an **Ambient temperature** T_{amb} (SI unit: K). The default is 293.15 K.

Periodic Heat Condition

Use the **Periodic Heat Condition** to add periodic heat conditions to boundary pairs. Right-click to add a **Destination Selection** subnode.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to apply the condition.

- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

Boundary Heat Source

The **Boundary Heat Source** models a heat source (or heat sink) that is embedded in the boundary. When selected as a **Pair Boundary Heat Source**, it also prescribes that the temperature field is continuous across the pair.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

When **Pair Boundary Heat Source** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

BOUNDARY HEAT SOURCE

Click the **General source** (the default) or **Total boundary power** button.

- If **General source** is selected, enter a value for the boundary heat source Q_{b} (SI unit: W/m^2) when the default option, **User defined**, is selected. A positive Q_{b} corresponds to heating and a negative Q_{b} corresponds to cooling.

The default is 0 W/m^2. For the general boundary heat source Q_b, there are predefined heat sources available when simulating heat transfer together with electrical or electromagnetic interfaces. Such sources represent, for example, ohmic heating and induction heating.

- If **Total boundary power** is selected, enter the total power (total heat source) $P_\mathrm{b,tot}$ (SI unit: W). The default is 0 W. In this case $Q_\mathrm{b} = P_\mathrm{b,tot}/A$, where A is the total area of the selected boundaries.

In 3D and 2D axial symmetry, $A = \int 1$.

In 2D and 1D axial symmetry:

$$A = dz\int 1$$

where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available to define dz.

In 1D:

$$A = A_c\int 1$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available to define A_c.

FRAME SELECTION

The settings are the same as for the Heat Source node and are described under the corresponding **Frame Selection** section.

- About Frames in Heat Transfer
- The Heat Transfer Interface

Continuity

The **Continuity** node can be added to pairs. It prescribes that the temperature field is continuous across the pair. **Continuity** is only suitable for pairs where the boundaries match.

BOUNDARY SELECTION

The selection list in this section shows the boundaries for the selected pairs.

PAIR SELECTION

When this node is selected from the **Pairs** menu, choose the pair on which to apply the condition. An identity pair has to be created first. Ctrl-click to deselect.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

Thin Thermally Resistive Layer

Use the **Thin Thermally Resistive Layer** node to define the thickness and thermal conductivity of a resistive material located on boundaries. It can be added to pairs by selecting **Pair Thin Thermally Resistive Layer** from the **Pairs** menu. The resistive material can also be defined through the **Thermal Resistance**:

$$R_s = \frac{d_s}{k_s}$$

The heat flux across the Thin Thermally Resistive Layer is defined by

$$-\mathbf{n}_d \cdot (-k_d \nabla T_d) = -k_s \frac{T_u - T_d}{d_s}$$

$$-\mathbf{n}_u \cdot (-k_u \nabla T_u) = -k_s \frac{T_d - T_u}{d_s}$$

where the u and d subscripts refer to the up- and downside of the layer, respectively.

When using the **Pair Thin Thermally Resistive Layer** node, then the u and d subscripts refer to the upside and the downside of the pair, respectively, instead of the layer.

Like any pair feature, the **Pair Thin Thermally Resistive Layer** condition contributes to any other pair feature. However, do not use two conditions of the same type on the same pair. In order to model a thin resistive layer made of several materials, use the **Multiple layers** option, which is available with the Heat Transfer Module.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on which to define the layer.

MODEL INPUTS

This section contains fields and values that are inputs to expressions defining material properties. If such user-defined materials are added, the model inputs appear here. Initially, this section is empty.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair on which to define the layer. An identity pair has to be created first. Ctrl-click to deselect.

THIN THERMALLY RESISTIVE LAYER

Select **Layer properties** (the default) or **Thermal resistance** from the **Specify** list.

- If **Layer properties** is selected, enter a value or expression for the **Layer thickness** d_{s} (SI unit: m). The default is 0.005 m. The default **Thermal conductivity** k_{s} (SI unit: W/(m·K)) is taken **From material**. Select **User defined** to enter another value or expression. The default is 0.01 W/(m·K).
- If **Thermal resistance** is selected, enter a value or expression for the **Thermal resistance** R_{s} (SI unit: s^3·K/kg). The default is s^3·K/kg.

Line Heat Source

The **Line Heat Source** node models a heat source (or sink) that is so thin that it has no thickness in the model geometry. Select this node from the **Edges** or **Points** submenu.

The **Line Heat Source** node is available in 3D on edges. In 2D and 2D axisymmetric it is available on points.

In theory, the temperature in a line source in 3D is plus or minus infinity (to compensate for the fact that the heat source does not have any volume). The finite element discretization used in COMSOL Multiphysics returns a finite temperature distribution along the line, but that distribution must be interpreted in a weak sense.

In 2D and 2D axisymmetric geometries the **Line Heat Source** is available on point level.

EDGE/POINT SELECTION

From the **Selection** list, choose the edges or points to define.

LINE HEAT SOURCE

Click the **General source** (the default) or **Total line power** button.

- When **General source** is selected, enter a value for the distributed heat source, Q_l (SI unit: W/m) in unit power per unit length. A positive Q_l corresponds to heating while a negative Q_l corresponds to cooling. The default is 0 W/m.
- If **Total line power** is selected, enter the total power (total heat source) $P_{\mathrm{l,tot}}$ (SI unit: W). The default is 0 W.

FRAME SELECTION

The settings are the same as for the Heat Source node and are described under the corresponding **Frame Selection** section.

- About Frames in Heat Transfer
- The Heat Transfer Interface

Point Heat Source

The **Point Heat Source** node models a heat source (or sink) that is so small that it can be considered to have no spatial extension. Select this node from the **Points** menu.

The **Point Heat Source** is available only in 3D

In theory, the temperature in a point source in 3D is plus or minus infinity (to compensate for the fact that the heat source does not have a spatial extension). The finite element discretization used in COMSOL Multiphysics returns a finite value, but that value must be interpreted in a weak sense.

POINT SELECTION

From the **Selection** list, choose the points on which to define the condition.

POINT HEAT SOURCE

Enter the **Point heat source** Q_p (SI unit: W) in unit power. A positive Q_p corresponds to heating while a negative Q_p corresponds to cooling. The default is 0 W.

Line Heat Source on Axis

The **Line Heat Source on Axis** node models a heat source (or sink) that is so thin that it has no thickness in the model geometry. Select this node from the **Edges** submenu.

The **Line Heat Source on Axis** is available only in 2D axisymmetric models.

EDGE SELECTION

From the **Selection** list, choose the edges on which to define the source. Note that the edges on the symmetry axis are the only applicable entities.

LINE HEAT SOURCE ON AXIS

The settings are the same as for the Line Heat Source node.

Point Heat Source on Axis

The **Point Heat Source on Axis** node models a heat source (or sink) that is so small that it can be considered to have no spatial extension. Select this node from the **Points** menu.

The **Point Heat Source on Axis** is available only in 2D axisymmetric models.

POINT SELECTION

From the **Selection** list, choose the points on which to apply the condition. Note that the points on the symmetry axis are the only applicable entities.

POINT HEAT SOURCE ON AXIS

The settings are the same for the Point Heat Source node.

The Joule Heating Interface

The **Joule Heating** ( ) multiphysics interface is used to model resistive heating and, depending on additional licensed products, dielectric heating in devices where inductive effects are negligible; that is, when the skin depth is much larger than the studied device. This multiphysics interface adds an Electric Currents interface and a Heat Transfer in Solids interface. The multiphysics couplings add the electromagnetic power dissipation as a heat source, and the electromagnetic material properties can depend on the temperature.

Depending on the licensed products, stationary modeling, time-domain modeling are supported in all space dimensions. In addition, combinations of frequency-domain modeling for the Electric Currents interface and stationary modeling for the Heat Transfer in Solids interface, called frequency-stationary and frequency-transient modeling, are supported.

When a predefined **Joule Heating** interface is added from the **Heat Transfer>Electromagnetic Heating** branch ( ) of the **Model Wizard** or **Add Physics** windows, **Electric Currents** and **Heat Transfer in Solids** interfaces are added to the Model Builder. In addition, The Multiphysics Node is added, which automatically includes the multiphysics coupling features **Electromagnetic Heat Source**, **Boundary Electromagnetic Heat Source**, and **Temperature Coupling**.

On the Constituent Physics Interfaces

The Electric Currents interface computes electric field, current and potential distributions in conducting media under conditions where inductive effects are negligible; that is, when the skin depth is much larger than the studied device. Depending on the licensed products, time and frequency domain formulations that account for capacitive effects are also provided. The Electric Currents interface solves a current conservation equation based on Ohm's law using the scalar electric potential as the dependent variable.

The Heat Transfer in Solids interface provides features for modeling heat transfer by conduction, convection, and radiation. A Heat Transfer in Solids model is active by default on all domains. All functionality for including other domain types, such as a fluid domain, is also available. The temperature equation defined in solid domains corresponds to the differential form of the Fourier's law that may contain additional contributions like heat sources.

In previous versions of COMSOL Multiphysics, a specific interface called Joule Heating was added to the Model Builder. Now, a predefined multiphysics coupling approach is used, improving the flexibility and design options for your modeling. For specific details, see Multiphysics Modeling Approaches and About This Release of COMSOL Multiphysics.

SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Joule Heating**, specific settings are included with the physics interfaces and the coupling features.

However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Electric Currents** and **Heat Transfer in Solids** interfaces are added, COMSOL adds an empty **Multiphysics** node. When you right-click this node, you can choose from the available coupling features,

Electromagnetic Heat Source, **Boundary Electromagnetic Heat Source**, and **Temperature Coupling**, but the modified settings are not included.

TABLE 14-5: MODIFIED SETTINGS FOR A JOULE HEATING INTERFACE

INTERFACE	MODIFIED SETTINGS (IF ANY)
Electric Currents	No changes.
Heat Transfer in Solids	No changes.
Electromagnetic Heat Source	The **Domain Selection** is the same as that of the participating physics interfaces. The corresponding **Electric Currents** and **Heat Transfer in Solids** interfaces are preselected in the Electromagnetic Heat Source section.
Boundary Electromagnetic Heat Source	The **Boundary Selection** contains all boundaries of the participating physics interfaces. The corresponding **Electric Currents** and **Heat Transfer in Solids** interfaces are preselected in the Boundary Electromagnetic Heat Source section.
Temperature Coupling	The corresponding **Electric Currents** and **Heat Transfer in Solids** interfaces are preselected in the Temperature Coupling section.

PHYSICS INTERFACE AND COUPLING FEATURES

Coupling Features

The Electromagnetic Heat Source, Boundary Electromagnetic Heat Source, and Temperature Coupling multiphysics coupling nodes are described in this section.

Physics Interface Features

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using.

- The available physics features for The Heat Transfer Interface are listed in the section Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer Interfaces.
- The available physics features for The Electric Currents Interface are listed in the section Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface.

If you have an add-on module, such as the Heat Transfer Module or AC/DC Module, there are additional specialized physics nodes available and described in the individual module documentation.

Joule Heating of a Microactuator—Distributed Parameter Version: model library path **COMSOL_Multiphysics/Multiphysics/thermal_actuator_simplified**

Electromagnetic Heat Source

The **Electromagnetic Heat Source** multiphysics coupling ( ) represents the source term Q (SI unit: W/m^3) in the heat equation implemented by

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \tag{14-22}$$

The resistive heating (ohmic heating) due to the electric current is

$$Q = \mathbf{J} \cdot \mathbf{E}$$

where $\mathbf{J}$ is the current density (SI unit: A/m^2), and $\mathbf{E}$ is the electric field strength (SI unit: V/m).

COUPLING IDENTIFIER

The coupling identifier is used primarily as a scope prefix for variables defined by a coupling node. Refer to such variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `emh`.

DOMAIN SELECTION

When nodes are added from the context menu, you can select **Manual** (the default) from the **Selection** list to choose specific domains to define the electromagnetic heat source or select **All domains** as required.

When **Electromagnetic Heat Source** is added as an effect of adding a **Joule Heating** or a **Joule Heating and Thermal Expansion** interface, the selection is the same as for the participating physics interfaces.

Only domains that are active in the physics interfaces selected in the **Electromagnetic Heat Source** section can be selected.

ELECTROMAGNETIC HEAT SOURCE

This section defines the physics involved in the electromagnetic heat source multiphysics coupling. By default, the applicable physics interface is selected in the **Electromagnetic** list to apply the **Heat transfer** to its physics interface to establish the coupling.

The default values depend on how the **Electromagnetic Heat Source** node is created.

- If it is added from the **Physics** ribbon (Windows users), **Physics** contextual toolbar (Mac and Linux users), or context menu (all users), then the first physics interface of each type in the component is selected as the default.
- If it is added automatically when a **Joule Heating** or a **Joule Heating and Thermal Expansion** interface is chosen in the **Model Wizard** or **Add Physics** window, then the two participating **Electric Currents** and **Heat Transfer in Solids** interfaces are selected.

You can also select **None** from either list to uncouple the **Electromagnetic Heat Source** node from a physics interface. If the physics interface is removed from the **Model Builder**, for example **Heat Transfer in Solids** is deleted, then the **Heat transfer** list defaults to **None** as there is nothing to couple to.

If a physics interface is deleted and then added to the model again, then in order to re-establish the coupling, you need to choose the physics interface again from the **Heat transfer** or **Electromagnetic** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present interface. See Multiphysics Modeling Approaches.

Boundary Electromagnetic Heat Source

The **Boundary Electromagnetic Heat Source** multiphysics coupling ( ) maps the electromagnetic surface losses as a heat source on the boundary (SI unit: W/m^2) in the heat transfer part of the model. It is a default node.

COUPLING IDENTIFIER

The coupling identifier is used primarily as a scope prefix for variables defined by a coupling node. Refer to such variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `bemh`.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

BOUNDARY ELECTROMAGNETIC HEAT SOURCE

This section defines the physics involved in the boundary electromagnetic heat source multiphysics coupling. By default, the applicable physics interface is selected in the **Electromagnetic** list to apply the **Heat transfer** to its physics interface to establish the coupling.

See the Electromagnetic Heat Source for more details about this section.

Temperature Coupling

Use the **Temperature Coupling** ( ) multiphysics coupling to add the temperature as the default model input for a standalone physics interface.

The **Temperature Coupling** feature is generic and specifies a Heat Transfer interface as **Source** and a second interface as **Destination**. When **Temperature Coupling** feature is used, the temperature from the **Source**, Heat Transfer interface is used to evaluate material properties in any feature from the **Destination** interface. The coupling can be added wherever the Heat Transfer interface is active.

The **Source** interface can be any interface defining a temperature, which includes all versions of heat transfer and multiphysics, except the pure radiation interfaces.

The **Destination** interface can be any interface providing multiphysics feature in the **Multiphysics** node, for example **Electric Current** or **Solid Mechanics**.

COUPLING IDENTIFIER

The coupling identifier is used primarily as a scope prefix for variables defined by a coupling node. Refer to such variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first coupling in the model) is `tc`.

TEMPERATURE COUPLING

This section defines the physics involved in the temperature coupling. By default, the software selects an appropriate physics interface for you from the **Source** and **Destination** lists. See the Electromagnetic Heat Source for more details about this section.

About Frames in Heat Transfer

This section discusses heat transfer analysis with moving frames, when spatial and material frames do not coincide. When the **Enable conversions between material and spatial frame** check box is selected, all heat transfer physics account for deformation effects on heat transfer properties.

The entire physics (equations and variables) are defined on the spatial frame. When a moving mesh is detected, the user inputs for certain features are defined on the material and are converted so that all the corresponding variables contain the value on the spatial frame.

- The Heat Transfer Interface
- Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer Interfaces
- Heat Transfer Theory

Frame Physics Feature Nodes and Definitions

This subsection contains the list of all heat transfer nodes and the corresponding definition frame.

Some of the physics require additional licenses, for example, a Heat Transfer Module or a CFD Module.

The following explains the different values listed in the *definition frame* column in Table 14-6, Table 14-7, and Table 14-8:

Material: The inputs are entered by the user and defined on the material frame. Because the heat transfer variables and equations are defined on the spatial frame, the inputs are internally converted to the spatial frame.

Spatial: The inputs are entered by the user and defined on the spatial frame. No conversion is done.

Material/(Spatial): For these physics nodes, select from a menu to decide if the inputs are defined on the material or spatial frame. The default definition frame is the material frame.

(Material)/Spatial: For these physics nodes, select from a menu to decide if the inputs are defined on the material or spatial frame. The default definition frame is the spatial frame.

N/A: There is no definition frame for this physics node.

Domain Nodes

TABLE 14-6: DOMAIN PHYSICS NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Bioheat	Material
Biological Tissue	Material
Change Thickness	Spatial
Geothermal Heating	Material
Heat Source	Material/(Spatial)

TABLE 14-6: DOMAIN PHYSICS NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Heat Transfer in Fluids	Spatial
Heat Transfer in Porous Media	Material (Solid part) Spatial (Fluid part)
Heat Transfer in Solids	Material
Heat Transfer with Phase Change	Spatial
Immobile Fluids	Spatial
Infinite Elements	Spatial
Initial Values	Spatial
Opaque	N/A
Out-of-Plane Convective Heat Flux	Spatial
Out-of-Plane Heat Flux	Spatial
Out-of-Plane Radiation	Spatial
Pressure Work	Spatial
Radiation in Participating Media	Spatial
Thermal Dispersion	Spatial
Translational Motion	Material
Viscous Heating	Spatial

Boundary Nodes

TABLE 14-7: BOUNDARY PHYSICS NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Boundary Heat Source	Material/(Spatial)
Continuity on Interior Boundary	Spatial
Convective Heat Flux	Spatial
Diffuse Mirror	Spatial
Heat Continuity	Spatial
Heat Flux	(Material)/Spatial
Highly Conductive Layer	Material
Inflow Heat Flux	Spatial
Layer Heat Source	Material
Opaque Surface	Spatial
Open Boundary	Spatial
Outflow	N/A
Pair Boundary Heat Source	Material/(Spatial)
Pair Thermal Contact	Material
Pair Thin Thermally Resistive Layer	Material
Periodic Heat Condition	Spatial
Prescribed Radiosity	Spatial
Radiation Group	N/A
Surface-to-Ambient Radiation	Spatial
Surface-to-Surface Radiation	Spatial
Symmetry	N/A
Temperature	Spatial
Thermal Contact	Material
Thermal Insulation	N/A
Thin Thermally Resistive Layer	Material

Edge and Point Nodes

TABLE 14-8: EDGE AND POINT NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Edge Heat Flux	(Material)/Spatial
Edge Surface-to-Ambient	Spatial
Line Heat Source	Material/(Spatial)
Point Heat Flux	Spatial
Point Heat Source	Material
Point Surface-to-Ambient	Spatial
Point Temperature	Spatial
Temperature	Spatial

TABLE 14-9: HEAT TRANSFER IN THIN SHELLS NODES

NODE NAME	DEFINITION FRAME
Change Effective Thickness	Spatial
Change Thickness	Spatial
Convective Heat Flux	Spatial
Edge Heat Source	Material/(Spatial)
Heat Flux	Spatial/(Material)
Heat Source	Material/(Spatial)
Initial Values	Spatial
Surface-to-Ambient Radiation	Spatial
Temperature	Spatial
Thin Conductive Layer	Material

Conversion Between Material and Spatial Frames

This subsection explains how the user inputs are converted. The conversion depends on the dimension of the variables (scalars, vectors, or tensors) and on their density order.

DENSITY, HEAT SOURCE, HEAT FLUX

Scalar density variables do not have the same value in the material and in the spatial frame.

In heat transfer physics, the following variables are relative scalars of weight one (also called scalar densities): the mass density ρ, the heat source Q, the heat flux q_0, the heat transfer coefficient h, and the production/absorption coefficient q_{s}.

When a feature has its *definition frame* on the spatial frame, no transformation is done because the user input is defined on the spatial frame. For example, if `ρ=500[kg/m^3]` is defined in the Heat Transfer in Fluids (*definition frame* = spatial frame) the variable `ht.rho` is equal to `500[kg/m^3]` (on the spatial frame).

When a feature has its *definition frame* on the material frame, the user input is defined on the material frame so it has to be multiplied by `spatial.detInvF` to get the corresponding value on the spatial frame. For example, if `ρ=500[kg/m^3]` is defined in the Heat Transfer in Solids (*definition frame* = material frame) the variable `ht.rho` is equal to `spatial.detInvF*500[kg/m^3]` (on the spatial frame). As a consequence, to evaluate or integrate the mass density on the material frame, the value of `ht.rho` has to be multiplied by `spatial.detF`.

`spatial.detF` has different definitions based on the dimension of the geometric entity where it is evaluated. On domains it corresponds to the local volume change from the material to the spatial frame while it corresponds to local surface or length change on boundaries and edges. `spatial.detInvF` is the inverse of `spatial.detF`.

VELOCITY VECTOR

The relationship between $\boldsymbol{u}_{(x,y,z)}$ and $\boldsymbol{u}_{(X,Y,Z)}$ is

$$\boldsymbol{u}_{(x,y,z)} = \boldsymbol{F}^T\boldsymbol{u}_{(X,Y,Z)}$$

where $\boldsymbol{F}$ is the coordinate transform matrix from the material to the spatial frame:

$$F = \begin{bmatrix} x_X & y_X & z_X \\ x_Y & y_Y & z_Y \\ x_Z & y_Z & z_Z \end{bmatrix}$$

with x_X corresponding to the derivative of x with respect to X.

THERMAL CONDUCTIVITY

Thermal conductivity is a tensor density. The relationship between the value on the spatial frame and the material frame is

$$\boldsymbol{k}_{(x, y, z)} = \frac{1}{\det(\boldsymbol{F})}\boldsymbol{F}^T\boldsymbol{k}_{(X, Y, Z)}\boldsymbol{F}$$

where $\boldsymbol{k}_{(x, y, z)}$ is the thermal conductivity tensor in the spatial frame and $\boldsymbol{k}_{(X, Y, Z)}$ is the thermal conductivity tensor in the material frame. $\boldsymbol{F}$ is the coordinate transform matrix from the material frame to the spatial frame defined in the paragraph above.

THERMAL CONDUCTIVITY OF HIGHLY CONDUCTIVE LAYER

The same transformations are applied to thermal conductivity but with different transformation matrices. The transformation matrix uses tangential derivatives and is defined as

$$F_{\text{tang}} = \begin{bmatrix} xT_X\; yT_X\; zT_X \\ xT_Y\; yT_Y\; zT_Y \\ xT_Z\; yT_Z\; zT_Z \end{bmatrix}$$

where xT_X corresponds to the tangential derivative x with respect to X, and so on.

AXISYMMETRIC GEOMETRIES

In 1D axisymmetric and 2D axisymmetric models an additional conversion is done between the material frame and the spatial frame. The density variables (density, heat source, heat flux, and so forth) are multiplied by

$$\frac{R}{r}$$

which corresponds to the ratio of the material first cylindrical coordinate over the spatial one.

For example, if you enter a heat source $\boldsymbol{Q}$ =`500[W/m^3]`in the material frame in axisymmetric cases, the conversion leads to:

$$\boldsymbol{Q} = \texttt{500[W/m^3]*}\frac{R}{r}\text{spatial.detInvF}$$

15

Structural Mechanics

This chapter explains how to use the Solid Mechanics interface, found under the **Structural Mechanics** branch ( ) when adding a physics interface, to simulate and analyze applications involving solid mechanics. The interface is used for stress analysis and general solid mechanics simulation.

The optional Structural Mechanics Module contains interfaces and models that allow for extended, specialized analyses of structural and solid mechanics problems.

Solid Mechanics Geometry and Structural Mechanics Physics Symbols

The Solid Mechanics interface in the Structural Mechanics Module is available for these space dimensions, which are described in this section:

- 3D Solid Geometry
- 2D Geometry (*plane stress* and *plane strain*)
- Axisymmetric Geometry

There are also physics symbols available with structural mechanics features as described in these sections:

- Physics Symbols for Boundary Conditions
- About Coordinate Systems and Physics Symbols
- Displaying Physics Symbols in the Graphics Window—An Example

3D Solid Geometry

The degrees of freedom (dependent variables) in 3D are the global displacements u, v, and w in the global x, y, and z directions, respectively, and the pressure help variable (used only if a nearly incompressible material is selected).

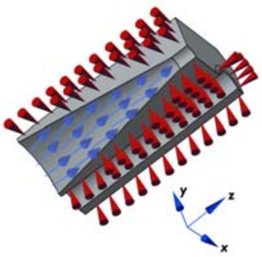


Figure 15-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.

2D Geometry

PLANE STRESS

The plane stress variant of the 2D interface is useful for analyzing thin in-plane loaded plates. For a state of plane stress, the out-of-plane components of the stress tensor are zero.

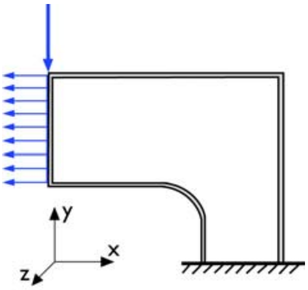


Figure 15-2: Plane stress models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

The 2D interface for plane stress allows loads in the x and y directions, and it assumes that these are constant throughout the material's thickness, which can vary with x and y. The plane stress condition prevails in a thin flat plate in the xy-plane loaded only in its own plane and without any z direction restraint.

PLANE STRAIN

The plane strain variant of the 2D interface that assumes that all out-of-plane strain components of the total strain ε_z, ε_{yz}, and ε_{xz} are zero.

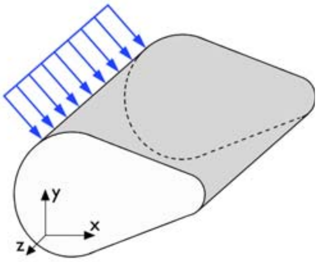


Figure 15-3: A geometry suitable for plane strain analysis.

Loads in the x and y directions are allowed. The loads are assumed to be constant throughout the thickness of the material, but the thickness can vary with x and y. The plane strain condition prevails in geometries, whose extent is large in the z direction compared to in the x and y directions, or when the z displacement is in some way restricted. One example is a long tunnel along the z-axis where it is sufficient to study a unit-depth slice in the xy-plane.

Axisymmetric Geometry

The axisymmetric variant of the Solid Mechanics interface uses cylindrical coordinates r, φ (`phi`), and z. Loads are independent of φ, and the axisymmetric variant of the interface allows loads only in the r and z directions.

The 2D axisymmetric geometry is viewed as the intersection between the original axially symmetric 3D solid and the half plane $\varphi = 0$, $r \geq 0$. Therefore the geometry is drawn only in the half plane $r \geq 0$ and recover the original 3D solid by rotating the 2D geometry about the z-axis.

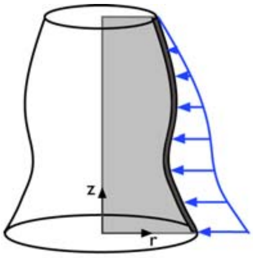


Figure 15-4: Rotating a 2D geometry to recover a 3D solid.

Physics Symbols for Boundary Conditions

To display the boundary condition symbols listed in Table 15-1, enable the **Show physics symbols** from the **Graphics and Plot Windows** menu on The Preferences Dialog Box. The check box is not selected by default.

These symbols are available with the applicable structural mechanics feature nodes.

TABLE 15-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY NODE	NOTES
	Added Mass[1]	Added Mass	
	Antisymmetry[1]	Antisymmetry	
	Body Load[1]	Body Load	
	3D Coordinate System		Green indicates the Y direction, blue indicates the Z direction, and red indicates the X direction.
	2D Coordinate System		Green indicates the Y direction and red indicates the X direction.
	Distributed Force	Boundary Load Face Load Edge Load	Can be displayed together with the Distributed Moment symbol, depending on the values given in the node.
	Damping[1]	Spring Foundation	Can be displayed together with the Spring symbol, depending on the values given in the node.
	Distributed Moment[1]	Boundary Load Face Load Edge Load	Can be displayed together with the Distributed Force symbol, depending on the values given in the node.
	Fixed Constraint	Fixed Constraint	
	No Rotation[1]	No Rotation	
	Pinned[1]	Pinned	
	Point Force	Point Load	Can be displayed together with the Point Moment symbol, depending on the values given in the node.
	Point Mass[1]	Point Mass	
	Point Moment[1]	Point Load	Can be displayed together with the Point Force symbol, depending on the values given in the node.
	Prescribed Acceleration	Prescribed Acceleration	
	Prescribed Displacement	Prescribed Displacement	
	Prescribed Velocity[1]	Prescribed Velocity	
	Prescribed Acceleration[1]	Prescribed Acceleration	
	Rigid Connector[1]	Rigid Connector	
	Roller	Roller	

TABLE 15-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY NODE	NOTES
	Spring[1]	Spring Foundation Thin Elastic Layer	Can be displayed together with the Damping symbol, depending on the values given in the node.
	Symmetry	Symmetry	
	Thin-Film Damping[2]	Thin-Film Damping	

[1] Requires the Structural Mechanics Module

[2] Requires the MEMS Module

About Coordinate Systems and Physics Symbols

Physics symbols connected to a node for which input can be given in different coordinate systems are shown together with a coordinate system symbol. This symbol is either a triad or a single arrow. The triad is shown if data are to be entered using vector components, as for a force. The single arrow is displayed when a scalar value, having an implied direction, is given. An example of the latter case is a pressure.

In both cases, the coordinate directions describe the direction in which a positive value acts. The coordinate direction symbols do not change with the values actually entered for the data.

Physics symbols are displayed even if no data values have been entered in the node.

For cases when physics symbol display is dependent on values actually given in the node, it can be necessary to move to another node before the display is actually updated on the screen.

Displaying Physics Symbols in the Graphics Window—An Example

1 To open the **Preferences** dialog box:
   - Windows users: From the **File** menu, select **Preferences** ( ) You can also customize the Quick Access Toolbar and then click the button.
   - Cross platform (Mac and Linux) users: From the main menu select **Options>Preferences** ( ).
2 Select **Graphics** and click the **Show physics symbols** check box. Click **OK**.
3 Add a physics interface, for example, **Solid Mechanics**, from the **Structural Mechanics** branch.

The physics symbols also display for any multiphysics interface that includes Structural Mechanics feature nodes.

4 Add any of the feature nodes listed in Table 15-1 to the interface. Availability is based on license and interface.

5 When adding the boundary, edge, or point (a geometric entity) to the **Selection** list in the feature settings window, the symbol displays in the **Graphics** window. See Figure 15-5.

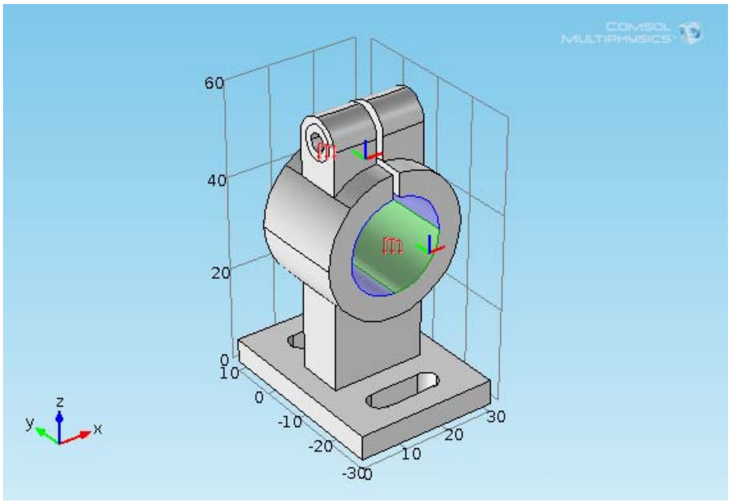
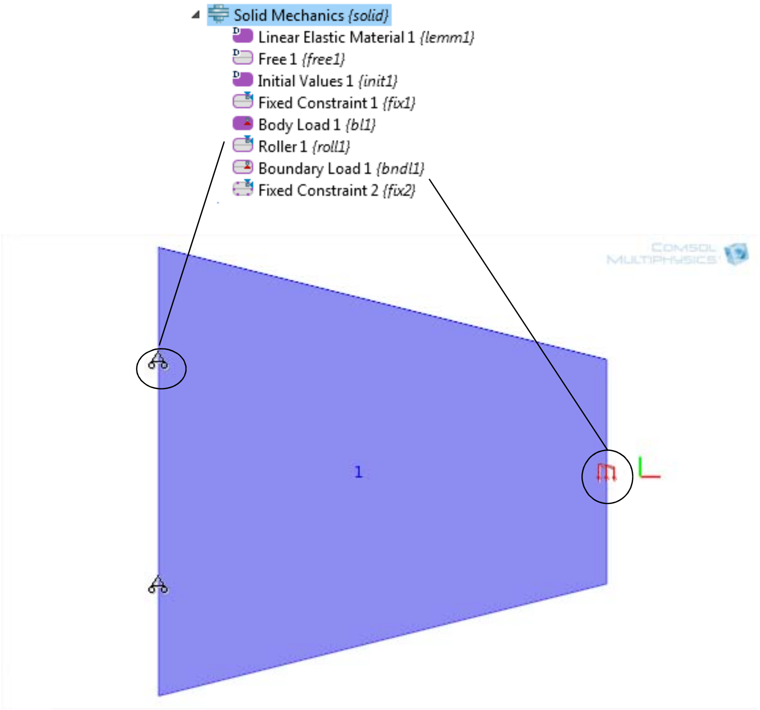


Figure 15-5: Example of the Boundary Load physics symbols as displayed in the COMSOL Multiphysics model library Deformation of a Feeder Clamp.

6 After assigning the boundary condition to a geometric entity, to display the symbol, click the top level physics interface node and view it in the **Graphics** window. See Figure 15-6.



- Solid Mechanics *{solid}*
  - Linear Elastic Material 1 *{lemm1}*
  - Free 1 *{free1}*
  - Initial Values 1 *{init1}*
  - Fixed Constraint 1 *{fix1}*
  - Body Load 1 *{bl1}*
  - Roller 1 *{roll1}*
  - Boundary Load 1 *{bndl1}*
  - Fixed Constraint 2 *{fix2}*

Figure 15-6: Example of Roller and Boundary Load physics symbols as displayed in the COMSOL Multiphysics model library “Tapered Cantilever”.

The Solid Mechanics Interface

The **Solid Mechanics (solid)** interface (), found under the **Structural Mechanics** branch () when adding a physics interface, is intended for general structural analysis of 3D, 2D, or axisymmetric bodies. In 2D, plane stress or plane strain assumptions can be used. The Solid Mechanics interface is based on solving Navier's equations, and results such as displacements, stresses, and strains are computed.

The Acoustics Module, MEMS Module, and Structural Mechanics Module add several features, for example geometric nonlinearity and advanced boundary conditions such as contact, follower loads, and non-reflecting boundaries.

With the Nonlinear Structural Materials Module or the Geomechanics Module, the interface is extended with, for example, material models for plasticity, hyperelasticity, creep, and concrete.

The **Linear Elastic Material** is the default material, which adds a linear elastic equation for the displacements and has a settings window to define the elastic material properties.

When this physics interface is added, these default nodes are also added to the **Model Builder**— **Linear Elastic Material**, **Free** (a boundary condition where boundaries are free, with no loads or constraints), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, solid mechanics material models, boundary conditions, and loads. You can also right-click **Solid Mechanics** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `solid`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the displacements and the equations that describe the solid mechanics. To choose specific domains, select **Manual** from the **Selection** list.

2D APPROXIMATION

From the **2D approximation** list select **Plane stress** or **Plane strain** (the default). For more information see the theory section.

When modeling using plane stress, the Solid Mechanics interface solves for the out-of-plane strain displacement derivative, $\frac{\partial w}{\partial Z}$, in addition to the displacement field $\mathbf{u}$.

THICKNESS

For 2D models, enter a value or expression for the **Thickness** d (SI unit: m). The default value of 1 m is suitable for plane strain models, where it represents a a unit-depth slice, for example. For plane stress models, enter the actual thickness, which should be small compared to the size of the plate for the plane stress assumption to be valid.

Use a Change Thickness node to change thickness in parts of the geometry if necessary.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasi-static**. Use **Quasi-static** to treat the elastic behavior as quasi-static (with no mass effects; that is, no second-order time derivatives). Selecting this option gives a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasi-static** is selected.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the coordinates for the **Reference point for moment computation** $\mathbf{x}_{\mathrm{ref}}$ (SI unit: m; variable `refpnt`). The resulting moments (applied or as reactions) are then computed relative to this reference point. During the results and analysis stage, the coordinates can be changed in the **Parameters** section in the result nodes.

DEPENDENT VARIABLES

The interface uses the global spatial components of the **Displacement field** u as dependent variables. You can change both the field name and the individual component names. If a new field name coincides with the name of another displacement field, the two fields (and the interfaces which define them) share degrees of freedom and dependent variable component names. You can use this behavior to connect a Solid Mechanics interface to a Shell directly attached to the boundaries of the solid domain, or to another Solid Mechanics interface sharing a common boundary.

A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select a **Displacement field**—**Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default). The **Frame** type in the Solid Mechanics interface is always **Material**.

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics
- Theory of Solid Mechanics

- Stresses in a Pulley: model library path **COMSOL_Multiphysics/Structural_Mechanics/stresses_in_pulley**
- Eigenvalue Analysis of a Crankshaft: model library path **COMSOL_Multiphysics/Structural_Mechanics/crankshaft**

Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics

The Solid Mechanics Interface has these domain, boundary, edge, point, and pair nodes and subnodes, listed in alphabetical order, are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users).

In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. However, to add subnodes, right-click the parent node.

Note that you can add force loads acting on all levels of the geometry for the physics interface. Add a:

- Body Load to domains (to model gravity effects, for example).
- Boundary Load to boundaries (a pressure acting on a boundary, for example).
- Edge Load to edges in 3D (a force distributed along an edge, for example).
- Point Load to points (concentrated forces at points).

If there are subsequent constraints specified on the same geometrical entity, the last one takes precedence.

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis
- For information about the **Perfectly Matched Layers** feature, see Infinite Element Domains and Perfectly Matched Layers.
- Continuity on Interior Boundaries
- Identity and Contact Pairs

- Body Load
- Boundary Load
- Change Thickness
- Damping
- Edge Load
- Fixed Constraint
- Free
- Initial Values
- Linear Elastic Material
- Periodic Condition
- Point Load
- Prescribed Displacement
- Roller

Linear Elastic Material

The **Linear Elastic Material** node adds the equations for a linear elastic solid and an interface for defining the elastic material properties. Right-click to add a Damping subnode.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define a linear elastic solid and compute the displacements, stresses, and strains, or select **All domains** as required.

MODEL INPUTS

Define model inputs, for example, the temperature field of the material uses a temperature-dependent material property. If no model inputs are required, this section is empty.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used when stresses or strains are presented in a local system.

LINEAR ELASTIC MATERIAL

Define the **Solid model** and the linear elastic material properties.

Solid Model

The **Solid model** is linear **Isotropic** elastic.

Density

The default **Density** ρ (SI unit: kg/m^3) uses values **From material**. If **User defined** is selected, enter another value or expression.

Specification of Elastic Properties for Isotropic Materials

For an **Isotropic Solid model**, from the **Specify** list select a pair of elastic properties for an isotropic material—**Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and shear-wave speeds**. For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

> Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another (see Table 15-2).

The individual property parameters are:

- **Young's modulus** (elastic modulus) E (SI unit: Pa). The default is 0 Pa.
- **Poisson's ratio** ν (dimensionless). The default is 0.
- **Shear modulus** G (SI unit: N/m^2. The default is 0 N/m^2.
- **Bulk modulus** K (SI unit: N/m^2). The default is 0 N/m^2.
- **Lamé parameter** λ (SI unit: N/m^2) and **Lamé parameter** μ (SI unit: N/m^2). The defaults are 0 N/m^2.
- **Pressure-wave speed** (longitudinal wave speed) c_p (SI unit: m/s). The default is 0 m/s.
- **Shear-wave speed** (transverse wave speed) c_s (SI unit: m/s). The default is 0 m/s.

> This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Change Thickness

Use the **Change Thickness** node for 2D models to model domains with a thickness other than the overall thickness defined in the physics interface's Thickness section.

DOMAIN SELECTION

From the **Selection** list, choose the domains to use a different thickness.

CHANGE THICKNESS

Enter a value for the **Thickness** d (SI unit: m). This value replaces the overall thickness for the domains selected above.

Damping

Right-click the Linear Elastic Material node to add a **Damping** subnode, which is used in Time Dependent, Eigenfrequency, and Frequency Domain studies to model damped problems. The node adds Rayleigh damping by default.

> The time-stepping algorithms also add numerical damping, which is independent of any explicit damping added.
>
> For the generalized alpha time-stepping algorithm it is possible to control the amount of numerical damping added.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node. Or select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

DAMPING SETTINGS

Rayleigh damping is the only available **Damping type**.

Enter the **Mass damping parameter** α_{dM} (SI unit: 1/s) and the **Stiffness damping parameter** β_{dK} (SI unit: s). The default values are 0 (no damping).

In this damping model, the damping parameter ξ is expressed in terms of the mass m and the stiffness k as

$$\xi = \alpha_{\mathrm{dM}} m + \beta_{\mathrm{dK}} k$$

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* α_{dM} and the stiffness damping parameter β_{dM}.

Initial Values

The **Initial Values** node adds initial values for the displacement field and structural velocity field that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. Add additional **Initial Values** nodes from the **Physics** toolbar.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter values or expressions for the initial values of the **Displacement field** $\mathbf{u}$ (SI unit: m) (the displacement components u, v, and w in 3D) (the default is 0 m), and the **Structural velocity field** $\partial\mathbf{u}/\partial t$ (SI unit: m/s) (the default is 0 m/s)).

Body Load

Add a **Body Load** to domains for modeling gravity or centrifugal loads, for example.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

Select a **Load type**—**Load defined as force per unit volume** (the default) or **Total force**. For 2D models, **Load defined as force per unit area** is also an option.

Then enter values or expressions for the components in the matrix based on the selection and the space dimension:

- **Body load** $\mathbf{F}_\mathrm{V}$ (SI unit: N/m^3)

- **Total force** $\mathbf{F}_{\mathrm{tot}}$ (SI unit: N). For total force, COMSOL Multiphysics divides the total force by the volume of the domains where the load is active.
- *For 2D models:* **Load** $\mathbf{F}_{\mathrm{A}}$ (SI unit: N/m^2). The body load as force per unit volume is then the value of F divided by the thickness.

Free

The **Free** node is the default boundary condition. It means that there are no constraints and no loads acting on the boundary.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically set up and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Boundary Load

Add a **Boundary Load** to boundaries for a pressure acting on a boundary, for example.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

Select a **Load type**—**Load defined as force per unit area** (the default), **Pressure**, or **Total force**. For 2D models, **Load defined as force per unit length** is also an option.

> After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the Solid Mechanics interface with another physics interface, it is also possible to choose a predefined load from this list.

Then enter values or expressions for the components in the matrix based on the selection and the space dimension:

- **Load** $\mathbf{F}_{\mathrm{A}}$ (SI unit: N/m^2). The body load as force per unit volume is then the value of F divided by the thickness.
- *For 2D models:* **Load** $\mathbf{F}_{\mathrm{L}}$ (SI unit: N/m).

- **Total force** $\mathbf{F}_{\mathrm{tot}}$ (SI unit: N). For total force, COMSOL Multiphysics then divides the total force by the area of the surfaces where the load is active.
- **Pressure** p (SI unit: Pa), which can represent a pressure or another external pressure. The pressure is positive when directed toward the solid.

Fixed Constraint

The **Fixed Constraint** node adds a condition that makes the geometric entity fixed (fully constrained); that is, the displacements are zero in all directions. For domains, this condition is selected from the **More** submenu.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose, the geometric entity (domains, boundaries, edges, or points) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair use. An identity pair has to be created first.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Boundary Conditions

Prescribed Displacement

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in one or more directions to the geometric entity (domain, boundary, edge, or point). For domains, this condition is selected from the **More** submenu.

If a displacement is prescribed in one direction, this leaves the solid free to deform in the other directions.

You can also define more general displacements as a linear combination of the displacements in each direction.

- If a prescribed displacement is not activated in any direction, this is the same as a Free constraint.
- If a zero displacement is applied in all directions, this is the same as a Fixed Constraint.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. If you choose another, local coordinate system, the displacement components change accordingly.

PRESCRIBED DISPLACEMENT

Define the prescribed displacements using a **Standard notation** (the default) or a **General notation**.

Standard Notation

To define the displacements individually, click the **Standard notation** button.

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and for 3D models, **Prescribed in z direction** check boxes. Then enter a value or expression for u_0, v_0, and for 3D models, w_0 (SI unit: m). For 2D axisymmetric models, select one or both of the **Prescribed in r direction** and **Prescribed in z direction** check boxes. Then enter a value or expression for u_0 and w_0 (SI unit: m).

General Notation

Click the **General notation** to specify the displacements using a general notation that includes any linear combination of displacement components. For example, for 2D models, use the relationship

$$H\begin{bmatrix} u \\ v \end{bmatrix} = R$$

For **H matrix** H (dimensionless) select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and then enter values as required in the field or matrix. Enter values or expressions for the **R vector** R (SI unit: m)

For example, to achieve the condition $u = v$, use the settings

$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which force the domain to move only diagonally in the xy-plane.

CONSTRAINT SETTINGS

See Fixed Constraint for these settings.

Roller

The **Roller** node adds a roller constraint as the boundary condition; that is, the displacement is zero in the direction perpendicular (normal) to the boundary, but the boundary is free to move in the tangential direction. See Fixed Constraint for all the settings.

CONSTRAINT SETTINGS

See Fixed Constraint for these settings.

Periodic Condition

The **Periodic Condition** node adds a periodic boundary condition. This periodicity makes $u_i(x_0) = u_i(x_1)$ for a displacement u_i. You can control the direction that the periodic condition applies to. If the source and destination boundaries are rotated with respect to each other, this transformation is automatically performed, so that corresponding displacement components are connected.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. The software automatically identifies the boundaries as either source boundaries or destination boundaries.

This works fine for cases like opposing parallel boundaries. In other cases right-click the **Periodic Condition** node to add a **Destination Selection** subnode to control the destination. By default it contains the selection that COMSOL Multiphysics has identified.

In cases where the periodic boundary is split into several boundaries within the geometry, it might be necessary to apply separate periodic conditions to each pair of geometry boundaries.

PERIODICITY SETTINGS

Select a **Type of periodicity**—**Continuity** (the default), **Antiperiodicity** or **User defined**. If **User defined** is selected, select the **Periodic in u**, **Periodic in v** (for 3D and 2D models), and **Periodic in w** (for 3D and 2D axisymmetric models) check boxes as required. Then for each selection, choose the **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

CONSTRAINT SETTINGS

See Fixed Constraint for these settings.

- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

Edge Load

Add an **Edge Load** to 3D models for a force distributed along an edge, for example.

EDGE SELECTION

From the **Selection** list, choose the edges to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

Select a **Load type**—**Load defined as force per unit area** (the default) or **Total force**. Then enter values or expressions for the components in the matrix based on the selection:

- **Load** $\mathbf{F}_L$ (SI unit: N/m). When combining the Solid Mechanics interface with, for example, film damping, it is also possible to choose a predefined load from this list.
- **Total force** $\mathbf{F}_{\mathrm{tot}}$ (SI unit: N). COMSOL Multiphysics then divides the total force by the volume where the load is active.

Point Load

Add a **Point Load** to points for concentrated forces at points.

POINT SELECTION

From the **Selection** list, choose the points to define.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

Enter values or expressions for the components of the **Point load** $\mathbf{F}_\mathrm{p}$ (SI unit: N).

Using Load Cases

For linear stationary problems it can be of interest to see the solution for several different loads ***F*** (right-hand side of the basic PDE) on the same structure (a model where the geometry and materials are defined and do not change). Typically this is used for studies using linear combinations of different loads—*load cases*. It is then possible to solve for these load cases in a computationally efficient way because there is no need to reassemble the stiffness matrix. Varying constraints can also be part of a general load case definition, and COMSOL supports load cases that are combination of loads, with optional weights, and constraints.

- The Relationship Between Study Steps and Solver Configurations
- Stationary

Defining Load Groups and Constraint Groups

For boundary conditions that represent loads and constraints, as well as other loads and constraints such as body loads, you can define *load groups* and *constraint groups*, which contain the loads and constraints, respectively, that you want to use as parts of load cases. All loads and constraints for structural mechanics as well as boundary conditions such as heat flux (a load) and temperature (a constraint) in heat transfer support load groups and constraint groups. You can create load groups and constraint groups in two ways—from the **Global Definitions** node's context menu or a physics node context menu. Both methods add the node under **Global Definitions**.

Add a Load or Constraint Group from the Global Definitions Context Menu

Add a a **Load Group** ( ) or **Constraint Group** ( ) under **Global Definitions** to create groups to which you can later assign loads and constraints. If you group the nodes, the load and constraint groups display under the **Load and Constraint Groups** node ( ). See Figure 15-7.

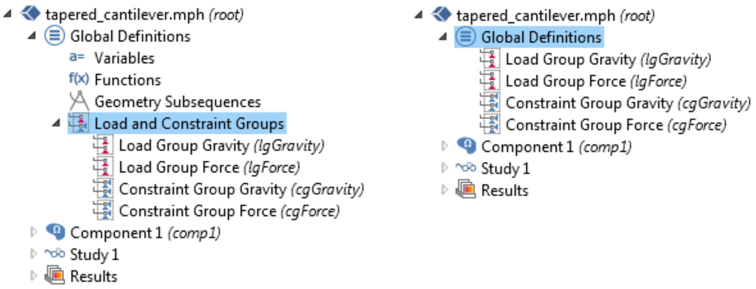

tapered_cantilever.mph (root)
Global Definitions
Variables
Functions
Geometry Subsequences
Load and Constraint Groups
Load Group Gravity (lgGravity)
Load Group Force (lgForce)
Constraint Group Gravity (cgGravity)
Constraint Group Force (cgForce)
Component 1 (comp1)
Study 1
Results
tapered_cantilever.mph (root)
Global Definitions
Load Group Gravity (lgGravity)
Load Group Force (lgForce)
Constraint Group Gravity (cgGravity)
Constraint Group Force (cgForce)
Component 1 (comp1)
Study 1
Results

Figure 15-7: An example of the node grouping when Group by Type is selected (left) and when Ungroup is selected (right).

Add a Load or Constraint Group from a Physics Node Context Menu

Right-click a physics node for any load or constraint (for example, a Fixed Constraint, Heat Source, or Boundary Load node) and choose **Load Group>New Load Group** or **Constraint Group>New Constraint Group**, respectively. In this case, the software creates a Load Group or Constraint Group under Global Definitions and at the same time assigns that physics node (a load or a constraint) to that group.

Assign a Load or Constraint to a Group

To assign a load or a constraint to a load group or constraint group, right-click the physics node for a load or constraint and from the **Load Group** or **Constraint Group** submenu choose one of the following (see Figure 15-8):

- **Active in All Load Groups** (or **Active in All Constraint Groups**). This is the default setting, which you can use for some boundary conditions or other parts of the physics that take part in all load cases.
- One of the defined load groups or constraint groups such as **Load Group 1**, **Load Group 2**, and so on.
- **New Load Group** (or **New Constraint Group**) to create a new group as described earlier in this section.

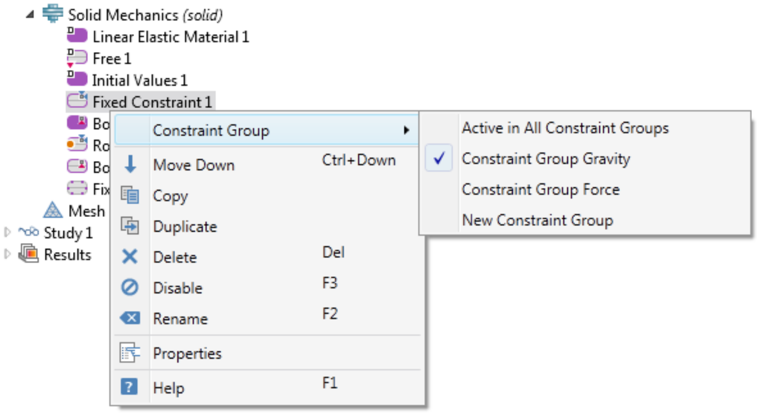
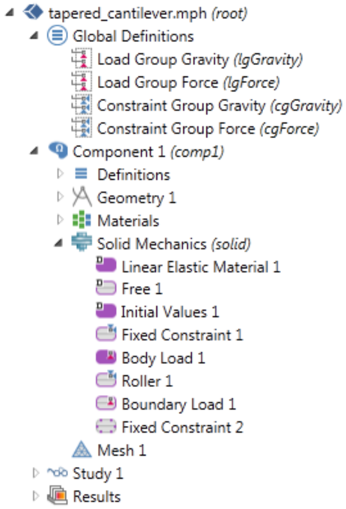


Figure 15-8: An example of the context menu options when a load or constraint physics node is right-clicked, in this case, the Fixed Constraint node.

Define as many groups as you need for the load cases that you want to study. Each load or constraint can only belong to one group. The next step is then to define the actual load cases as combinations of these groups (see Defining and Evaluating Load Cases).

When the Load Group or Constraint Group is applied to a node under a physics, the node indicates this visually. For example, the **Fixed Constraint** and **Roller** nodes have the blue **Constraint Group** symbol in the upper right corner and the **Body Load** and **Boundary Load** nodes have the red **Load Group** symbol in the upper right corner as in

Figure 15-9.



tapered_cantilever.mph *(root)*
Global Definitions
Load Group Gravity *(lgGravity)*
Load Group Force *(lgForce)*
Constraint Group Gravity *(cgGravity)*
Constraint Group Force *(cgForce)*
Component 1 *(comp1)*
Definitions
Geometry 1
Materials
Solid Mechanics *(solid)*
Linear Elastic Material 1
Free 1
Initial Values 1
Fixed Constraint 1
Body Load 1
Roller 1
Boundary Load 1
Fixed Constraint 2
Mesh 1
Study 1
Results

Figure 15-9: An example of the Load and Constraint Groups defined under Global Definitions with loads and constraints applied to nodes under Solid Mechanics.

- Tapered Cantilever with Two Load Cases: model library path **COMSOL_Multiphysics/Structural_Mechanics/tapered_cantilever**
- If you have the Structural Mechanics Module, also see Pratt Truss Bridge: model library path **Structural_Mechanics_Module/Civil_Engineering/pratt_truss_bridge**.

Load Group

Add a **Load Group** ( ) to the **Global Definitions** branch to create a load group to which you can assign one or more loads. You can then activate the load group in one or more *load cases* for efficiently solving a structural mechanics or heat transfer model to analyze the effects of various loads or sources.

GROUP IDENTIFIER

In the **Identifier** field, the default group identifier (`lg1`) can be modified if you want to use a more descriptive name, for example, `lgGravity`. You can also create load groups from physics nodes for structural mechanics that represent loads and support load cases: Right-click the physics node and choose **Load Group>New Load Group**.

Constraint Group

Add a **Constraint Group** ( ) to the **Global Definitions** branch to create a constraint group to which you can assign one or more constraints. You can then activate the constraint group in one or more *load cases* for efficiently solving a model to analyze the effects of various constraints.

GROUP IDENTIFIER

In the **Identifier** field you can modify the default group identifier (`cg1`) if you want to use a more descriptive name (for example, `cgForce`). You can also create constraint groups from physics nodes for structural mechanics that represent constraints and support load cases: Right-click the physics node and choose **Constraint Group>New Constraint Group**.

If you select **Group by Type** from the context menu, either right-click the **Global Definitions** node or the **Loads and Constraints Groups** node ( ) to add a **Load Group** or **Constraint Group**.

Defining and Evaluating Load Cases

You define load cases in the **Stationary** study node's settings window. Follow these steps to create load cases:

1 In the **Model Builder** under **Study**, in the **Stationary** study ( ) settings window click to expand the **Study Extensions** section.

2 Select the **Define load cases** check box (see Figure 15-10).

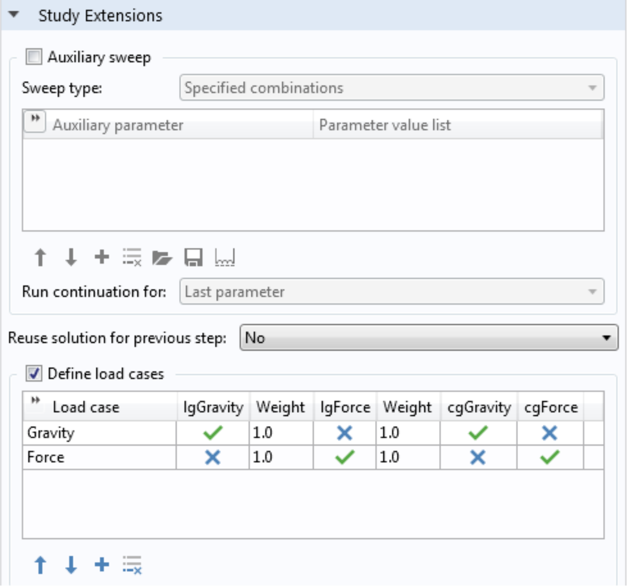
3 In the **Define load cases** area, click the **Add** (+) button underneath the table to add a load case.

4 The added load case appears last in the table of load cases. Use the **Move Up** (↑), **Move Down** (↓), and **Delete** ( ) buttons to rearrange the load cases in the table, and click the **Add** (+) button to add more load cases.

5 For each load case you can change its name from the default (Load case 2, for example) in the **Load case** column.

6 Include the load groups and constraint groups for each load case by clicking the ✕ in the columns for the groups to include. The symbol then changes to ✓ in order to indicate that the group participates in the load case.

7 For load groups, optionally change the weight from its default value of 1.0 in the corresponding **Weight** column (the **Weight** column to the right of the load group that it affects). Use a positive value other than 1 to increase or decrease the magnitude of the load; a negative value also reverses the load's direction.



Study Extensions

☐ Auxiliary sweep

Sweep type: Specified combinations

Auxiliary parameter	Parameter value list

Run continuation for: Last parameter

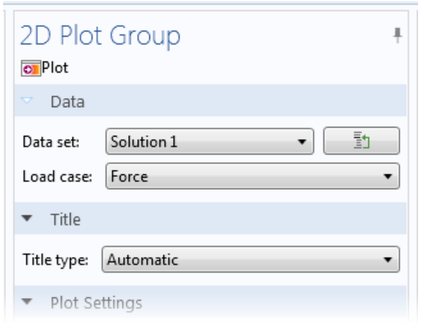
Reuse solution for previous step: No

☑ Define load cases

Load case	lgGravity	Weight	lgForce	Weight	cgGravity	cgForce
Gravity	✓	1.0	✕	1.0	✓	✕
Force	✕	1.0	✓	1.0	✕	✓

Figure 15-10: An example of the Stationary node's Study Extensions section. Click in the table of load cases to select or remove loads and constraints from the load case.

When you have defined all load cases you can compute the solution. COMSOL then solves for all load cases directly. In the plot groups that are created, a **Load case** list in the **Data** section of the **Plot Group** settings windows contains all load cases. To plot using the solution for a specific load case, select the load case of interest from the **Load case** list, and then click **Plot** ( ).



Data set: Solution 1

Load case: Force

Title type: Automatic

- Introduction to Solvers and Studies
- Stationary study step

Theory of Solid Mechanics

Material and Spatial Coordinates

The Solid Mechanics interface, through its equations, describes the motion and deformation of solid objects in a 2- or 3-dimensional space. In COMSOL Multiphysics terminology, this physical space is known as the *spatial frame* and positions in the physical space are identified by lowercase spatial coordinate variables x, y, and z (or r, φ , and z in axisymmetric models).

Continuum mechanics theory also makes use of a second set of coordinates, known as *material* (or *reference*) coordinates. These are normally denoted by uppercase variables X, Y, and Z (or R, Φ, and Z) and are used to label material particles. Any material particle is uniquely identified by its position in some given initial or reference configuration. As long as the solid stays in this configuration, material and spatial coordinates of every particle coincide and displacements are zero by definition.

When the solid objects deform due to external or internal forces and constraints, each material particle keeps its material coordinates $\mathbf{X}$ (bold font is used to denote coordinate vectors), while its spatial coordinates change with time and applied forces such that it follows a path

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \tag{15-1}$$

in space. Because the material coordinates are constant, the current spatial position is uniquely determined by the displacement vector $\mathbf{u}$, pointing from the reference position to the current position. The global Cartesian components of this displacement vector in the spatial frame, by default called u, v, and w, are the primary dependent variables in the Solid Mechanics interface.

By default, the Solid Mechanics interface uses the calculated displacement and Equation 15-1 to define the difference between spatial coordinates $\mathbf{x}$ and material coordinates $\mathbf{X}$. This means the material coordinates relate to the original geometry, while the spatial coordinates are solution dependent.

Material coordinate variables X, Y, and Z must be used in coordinate-dependent expressions that refer to positions in the original geometry, for example, for material properties that are supposed to follow the material during deformation. On the other hand, quantities that have a coordinate dependence in physical space, for example, a spatially varying electromagnetic field acting as a force on the solid, must be described using spatial coordinate variables x, y, and z.

Coordinate Systems

Force vectors, stress and strain tensors, as well as various material tensors are represented by their components in a specified coordinate system. By default, material properties use the canonical system in the material frame. This is the system whose basis vectors coincide with the X, Y, and Z axes. When the solid deforms, these vectors rotate with the material.

Loads and constraints, on the other hand, are applied in spatial directions, by default in the canonical spatial coordinate system. This system has basis vectors in the x, y, and z directions, which are forever fixed in space. Both the material and spatial default coordinate system are referred to as the *global coordinate system* in the physics interface.

Vector and tensor quantities defined in the global coordinate system on either frame use the frame’s coordinate variable names as indices in the tensor component variable names.

It is possible to define any number of user coordinate systems on the material and spatial frames. Most types of coordinate systems are specified only as a rotation of the basis with respect to the canonical basis in an underlying frame. This means that they can be used both in contexts requiring a material system and in contexts requiring a spatial one.

The coordinate system can be selected separately for each added material model, load, and constraint. This is convenient if, for example, an anisotropic material with different orientation in different domains is required. The currently selected coordinate system is known as the *local coordinate system*.

Lagrangian Formulation

The formulation used for structural analysis in COMSOL Multiphysics for both small and finite deformations is total Lagrangian. This means that the computed stress and deformation state is always referred to the material configuration, rather than to current position in space.

Likewise, material properties are always given for material particles and with tensor components referring to a coordinate system based on the material frame. This has the obvious advantage that spatially varying material properties can be evaluated just once for the initial material configuration and do not change as the solid deforms and rotates.

The gradient of the displacement, which occurs frequently in the following theory, is always computed with respect to material coordinates. In 3D:

$$\nabla\mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & \frac{\partial w}{\partial Z} \end{bmatrix}$$

The displacement is considered as a function of the material coordinates (X, Y, Z), but it is not explicitly a function of the spatial coordinates (x, y, z). It is thus only possible to compute derivatives with respect to the material coordinates.

About Linear Elastic Materials

The total strain tensor is written in terms of the displacement gradient

$$\varepsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$$

or in components as

$$\varepsilon_{mn} = \frac{1}{2}\left(\frac{\partial u_m}{\partial x_n} + \frac{\partial u_n}{\partial x_m}\right) \qquad (15\text{-}2)$$

The Hooke's law relates the stress tensor to the strain tensor and temperature:

$$s = s_0 + \mathsf{C} : (\varepsilon - \varepsilon_0 - \varepsilon_{\mathrm{th}}) \qquad (15\text{-}3)$$

where C is the 4th order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction), s_0 and ε_0 are initial stresses and strains, $\varepsilon_{\mathrm{th}} = \alpha(T - T_{\mathrm{ref}})$ is the thermal strain, and α is the coefficient of thermal expansion.

The elastic strain energy density is

$$W_s = \frac{1}{2}(\varepsilon - \varepsilon_0 - \varepsilon_{\text{th}}) : \mathsf{C} : (\varepsilon - \varepsilon_0 - \varepsilon_{\text{th}}) \tag{15-4}$$

or using the tensor components:

$$W_s = \sum_{i,j,m,n} \frac{1}{2}\mathsf{C}^{ijmn}(\varepsilon_{ij} - \varepsilon_{ij}^{0} - \varepsilon_{ij}^{\text{th}})(\varepsilon_{mn} - \varepsilon_{mn}^{0} - \varepsilon_{mn}^{\text{th}})$$

TENSOR VS. MATRIX FORMULATIONS

Because of the symmetry, the strain tensor can be written as the following matrix:

$$\begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix}$$

Similar representation applies to the stress and the thermal expansion tensors:

$$\begin{bmatrix} s_x & s_{xy} & s_{xz} \\ s_{xy} & s_y & s_{yz} \\ s_{xz} & s_{yz} & s_z \end{bmatrix}, \begin{bmatrix} \alpha_x & \alpha_{xy} & \alpha_{xz} \\ \alpha_{xy} & \alpha_y & \alpha_{yz} \\ \alpha_{xz} & \alpha_{yz} & \alpha_z \end{bmatrix}$$

Due to the symmetry, the elasticity tensor can be completely represented by a symmetric 6-by-6 matrix as:

$$D = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\ D_{12} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{13} & D_{23} & D_{33} & D_{34} & D_{35} & D_{36} \\ D_{14} & D_{24} & D_{34} & D_{44} & D_{45} & D_{46} \\ D_{15} & D_{25} & D_{35} & D_{45} & D_{55} & D_{56} \\ D_{16} & D_{26} & D_{36} & D_{46} & D_{56} & D_{66} \end{bmatrix} = \begin{bmatrix} \mathsf{C}^{1111} & \mathsf{C}^{1122} & \mathsf{C}^{1133} & \mathsf{C}^{1112} & \mathsf{C}^{1123} & \mathsf{C}^{1113} \\ \mathsf{C}^{1122} & \mathsf{C}^{2222} & \mathsf{C}^{2233} & \mathsf{C}^{2212} & \mathsf{C}^{2223} & \mathsf{C}^{2213} \\ \mathsf{C}^{1133} & \mathsf{C}^{2233} & \mathsf{C}^{3333} & \mathsf{C}^{3312} & \mathsf{C}^{3323} & \mathsf{C}^{3313} \\ \mathsf{C}^{1112} & \mathsf{C}^{2212} & \mathsf{C}^{3312} & \mathsf{C}^{1212} & \mathsf{C}^{1223} & \mathsf{C}^{1213} \\ \mathsf{C}^{1123} & \mathsf{C}^{2223} & \mathsf{C}^{3323} & \mathsf{C}^{1223} & \mathsf{C}^{2323} & \mathsf{C}^{2313} \\ \mathsf{C}^{1113} & \mathsf{C}^{2213} & \mathsf{C}^{3313} & \mathsf{C}^{1213} & \mathsf{C}^{2313} & \mathsf{C}^{1313} \end{bmatrix}$$

which is the *elasticity matrix.*

ISOTROPIC MATERIAL AND ELASTIC MODULI

In this case, the elasticity matrix becomes

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

and the thermal expansion is the diagonal tensor:

$$\begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix}$$

Different pairs of elastic moduli can be used, and as long as two moduli are defined, the others can be computed according to Table 15-2.

TABLE 15-2: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(K,G)$	$D(\lambda,\mu)$
Young's modulus	E		$\dfrac{9KG}{3K+G}$	$\mu\dfrac{3\lambda+2\mu}{\lambda+\mu}$
Poisson's ratio	ν		$\dfrac{1}{2}\left(1-\dfrac{3G}{3K+G}\right)$	$\dfrac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	K	$\dfrac{E}{3(1-2\nu)}$		$\lambda+\dfrac{2\mu}{3}$
Shear modulus	G	$\dfrac{E}{2(1+\nu)}$		μ
Lamé parameter λ	λ	$\dfrac{E\nu}{(1+\nu)(1-2\nu)}$	$K-\dfrac{2G}{3}$	
Lamé parameter μ	μ	$\dfrac{E}{2(1+\nu)}$	G	
Pressure-wave speed	c_p		$\sqrt{\dfrac{K+4G/3}{\rho}}$	
Shear-wave speed	c_s		$\sqrt{G/\rho}$	

According to Table 15-2, the elasticity matrix D for isotropic materials is written in terms of Lamé parameters λ and μ,

$$D = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

or in terms of the bulk modulus K and shear modulus G:

$$D = \begin{bmatrix} K+\frac{4G}{3} & K-\frac{2G}{3} & K-\frac{2G}{3} & 0 & 0 & 0 \\ K-\frac{2G}{3} & K+\frac{4G}{3} & K-\frac{2G}{3} & 0 & 0 & 0 \\ K-\frac{2G}{3} & K-\frac{2G}{3} & K+\frac{4G}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix}$$

Strain-Displacement Relationship

The strain conditions at a point are completely defined by the deformation components—u, v, and w in 3D—and their derivatives. The precise relation between strain and deformation depends on the relative magnitude of the displacement.

SMALL DISPLACEMENTS

Under the assumption of small displacements, the normal strain components and the shear strain components are related to the deformation as follows:

$$\begin{aligned}
\varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_{xy} &= \frac{\gamma_{xy}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\varepsilon_y &= \frac{\partial v}{\partial y} & \varepsilon_{yz} &= \frac{\gamma_{yz}}{2} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
\varepsilon_z &= \frac{\partial w}{\partial z} & \varepsilon_{xz} &= \frac{\gamma_{xz}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right).
\end{aligned} \tag{15-5}$$

To express the shear strain, use either the tensor form, ε_{xy}, ε_{yz}, ε_{xz}, or the engineering form, γ_{xy}, γ_{yz}, γ_{xz}.

The symmetric strain tensor ε consists of both normal and shear strain components:

$$\varepsilon = \begin{bmatrix} \varepsilon_x & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_y & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_z \end{bmatrix}$$

The strain-displacement relationships for the axial symmetry case for small displacements are

$$\varepsilon_r = \frac{\partial u}{\partial r}, \qquad \varepsilon_\varphi = \frac{u}{r}, \qquad \varepsilon_z = \frac{\partial w}{\partial z}, \text{ and} \qquad \gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}$$

Stress-Strain Relationship

The symmetric stress tensor σ describes stress in a material:

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \qquad \tau_{xy} = \tau_{yx} \qquad \tau_{xz} = \tau_{zx} \qquad \tau_{yz} = \tau_{zy}$$

This tensor consists of three normal stresses (σ_x, σ_y, σ_z) and six (or, if symmetry is used, three) shear stresses (τ_{xy}, τ_{yz}, τ_{xz}).

Plane Strain and Plane Stress Cases

For a general anisotropic linear elastic material in case of plane stress, COMSOL Multiphysics solves three equations $s_{i3} = 0$ for ε_{i3} with $i = 1, 2, 3$, and uses the solution instead of Equation 15-2 for these three strain components. Thus, three components ε_{i3} are treated as extra degrees of freedom. For isotropy and orthotropy, only with an extra degree of freedom, ε_{33},is used since all out of plane shear components of both stress and strain are zero. The remaining three strain components are computed as in 3D case according to Equation 15-2.

For an isotropic material, only the normal out-of-plane component ε_{33} needs to be solved for.

In case of plane strain, set $\varepsilon_{i3} = 0$ for $i = 1, 2, 3$. The out-of-plane stress components s_{i3} are results and analysis variables.

Axial Symmetry

The axially symmetric geometry uses a cylindrical coordinate system. Such a coordinate system is orthogonal but curvilinear, and one can choose between a covariant basis $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ and a contravariant basis $\mathbf{e}^1$, $\mathbf{e}^2$, $\mathbf{e}^3$.

The metric tensor is

$$[g_{ij}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in the coordinate system given by $\mathbf{e}^1$, $\mathbf{e}^2$, $\mathbf{e}^3$, and

$$\left[g^{ij} \right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^{-2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$.

The metric tensor plays the role of a unit tensor for a curvilinear coordinate system.

For any vector or tensor A, the metric tensor can be used for conversion between covariant, contravariant, and mixed components:

$$A_i^j = \sum_m (A_{im} g^{mj})$$

$$A^{ij} = \sum_{m,\,n} (A_{nm} g^{ni} g^{mj})$$

In both covariant and contravariant basis, the base vector in the azimuthal direction has a nonunit length. To cope with this issue, the so called physical basis vectors of unit length are introduced. These are

$$\mathbf{e}_r = \mathbf{e}_1 = \mathbf{e}^1, \mathbf{e}_\varphi = \frac{1}{r}\mathbf{e}_2 = r\mathbf{e}_2, \mathbf{e}_z = \mathbf{e}_3 = \mathbf{e}^3$$

The corresponding components for any vector or tensor are called physical.

For any tensor, the physical components are defined as

$$A_{ij}^{\text{phys}} = \sqrt{g_{ii}}\sqrt{g_{jj}}A^{ij}$$

where no summation is done over repeated indices.

DISPLACEMENTS AND AXIAL SYMMETRY ASSUMPTIONS

The axial symmetry implementation in COMSOL Multiphysics assumes independence of the angle, and also that the torsional component of the displacement is identically zero. The physical components of the radial and axial displacement, u and w, are used as dependent variables for the axially symmetric geometry.

For the linear elastic material, the stress components in coordinate system are

$$s^{ij} = s_0^{ij} + \mathrm{C}^{ijkl}(\varepsilon_{kl} - \alpha_{kl}\theta - \varepsilon_{0kl})$$

where $\theta = T - T_{\text{ref}}$.

For an isotropic material:

$$C^{ijkl} = \lambda g^{ij} g^{kl} + \mu(g^{ik} g^{jl} + g^{il} g^{jk})$$

where λ and μ are the first and second Lamé elastic parameters.

Loads

Specify loads as

- Distributed loads. The load is a distributed force in a volume, on a face, or along an edge.
- Total force. The specification of the load is as the total force. The software then divides this value with the area or the volume where the force acts.
- Pressure (boundaries only).

For 2D models choose how to specify the distributed boundary load as a load defined as force per unit area or a load defined as force per unit length acting on boundaries.

In the same way, choose between defining the load as force per unit volume or force per unit area for body loads acting in a domain. Also define a total force (SI unit: N) as required.

For 2D and axisymmetric models, the boundary loads apply on edges (boundaries).

For 2D axisymmetric models, the boundary loads apply on edges (boundaries).

For 3D solids, the boundary loads apply on faces (boundaries).

Table 15-3 shows how to define distributed loads on different geometric entity levels; the entries show the SI unit in parentheses.

TABLE 15-3: DISTRIBUTED LOADS

GEOMETRIC ENTITY	POINT	EDGE	FACE	DOMAIN
2D	force (N)	force/area (N/m^2) or force/length (N/m)	Not available	force/volume (N/m^3) or force/area (N/m^2)
Axial symmetry	total force along the circumferential (N)	force/area (N/m^2)	Not available	force/volume (N/m^3)
3D	force (N)	force/length (N/m)	force/area (N/m^2)	force/volume (N/m^3)

Pressure Loads

A pressure load is directed inward along the normal of boundary on which it is acting. This load type acts as a source of nonlinearity, since its direction depends on the current direction of the boundary normal. In a linearized context, for example in the frequency domain, the pressure is equivalent to a specified normal stress.

For general cases, if the problem is linear in all other respects, the solution can be made more efficient by forcing the solver to treat the problem as linear. See Stationary Solver.

Equation Implementation

The COMSOL Multiphysics implementation of the equations in the Solid Mechanics interface is based on the *principle of virtual work*.

The principle of virtual work states that the sum of virtual work from internal strains is equal to work from external loads.

The total stored energy, W, for a linear material from external and internal strains and loads equals:

$$W = \int_V (-\varepsilon : s + \mathbf{u} \cdot \mathbf{F}_V)dv$$

$$+ \int_S (\mathbf{u} \cdot \mathbf{F}_S)ds + \int_L (\mathbf{u} \cdot \mathbf{F}_L)dl + \sum_p (\mathbf{U}^t \cdot \mathbf{F}_p)$$

The principle of virtual work states that $\delta W = 0$ which leads to

$$\int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho\mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt})dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S)ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L)dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p)$$

Setting up Equations for Different Studies

The Solid Mechanics interface supports Stationary (static), Eigenfrequency, Time Dependent (transient), and Modal solver study types.

STATIONARY STUDIES

COMSOL Multiphysics uses an implementation based on the stress and strain variables. The normal and shear strain variables depend on the displacement derivatives.

Using the tensor strain, stress, and displacement variables, the principle of virtual work is expressed as:

$$\delta W = \int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V)dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S)ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L)dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p)$$

TIME DEPENDENT STUDIES

$$\int_V (-\varepsilon_{\text{test}} : (s + \beta_{dM} s_t) + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt} - \alpha_{dM} \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_t) dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p) \tag{15-6}$$

where the terms proportional to α_{dM} and β_{dK} appear if the *Rayleigh damping* is used

FREQUENCY-DOMAIN STUDIES

In the frequency domain the frequency response is studied when applying harmonic loads. Harmonic loads are specified using two components:

- The amplitude value, F_x
- The phase, $F_{x\text{Ph}}$

To derive the equations for the linear response from harmonic excitation loads

$$F_{x\text{freq}} = F_x(f) \cos\left(\omega t + F_{x\text{Ph}}(f) \frac{\pi}{180}\right)$$

$$\mathbf{F}_{\text{freq}} = \begin{bmatrix} F_{x\text{freq}} \\ F_{y\text{freq}} \\ F_{z\text{freq}} \end{bmatrix}$$

assume a harmonic response with the same angular frequency as the excitation load

$$u = u_{\text{amp}} \cos(\omega t + \phi_u)$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Also describe this relationship using complex notation

$$u = \text{Re}(u_{\text{amp}} e^{j\phi_u} e^{j\omega t}) = \text{Re}(\tilde{u} e^{j\omega t}) \text{ where } \tilde{u} = u_{\text{amp}} e^{j\phi_u}$$

$$\mathbf{u} = \text{Re}(\tilde{\mathbf{u}} e^{j\omega t})$$

$$F_{x\text{freq}} = \text{Re}\left(F_x(\omega) e^{jF_{xPh}(f)\frac{\pi}{180}} e^{j\omega t}\right) = \text{Re}(\tilde{F}_x e^{j\omega t})$$

where

$$\tilde{F}_x = F_x(f) e^{jF_{xPh}(f)\frac{\pi}{180}}$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{bmatrix}$$

EIGENFREQUENCY STUDIES

The eigenfrequency equations are derived by assuming a harmonic displacement field, similar as for the frequency response formulation. The difference is that this study type uses a new variable $j\omega$ explicitly expressed in the eigenvalue $j\omega = -\lambda$. The eigenfrequency f is then derived from $j\omega$ as

$$f = \left|\frac{\mathrm{Im}(j\omega)}{2\pi}\right|$$

Damped eigenfrequencies can be studied by adding viscous damping terms to the equation. In addition to the eigenfrequency the quality factor, Q, and decay factor, δ, for the model can be examined:

$$Q = \frac{\mathrm{Im}(\lambda)}{2\mathrm{Re}(\lambda)}$$

$$\delta = \mathrm{Re}(\lambda)$$

Damping Models

The Solid Mechanics interface offers two predefined damping models: Rayleigh damping and loss factor damping.

RAYLEIGH DAMPING

To model damping effects within the material, COMSOL Multiphysics uses *Rayleigh damping*, where two damping coefficients are specified.

The weak contribution due to the alpha-damping is always accounted for as shown in Equation 15-2. The contribution from the beta-damping that shown in Equation 15-6 corresponds to the case of small strains. In case of geometric nonlinearity, it becomes

$$\int_V (-\beta_{dM} \nabla \mathbf{u}_{\mathrm{test}} : P_t) dv$$

where P is the first Piola-Kirchhoff stress tensor.

To further clarify the use of the Rayleigh damping, consider a system with a single degree of freedom. The equation of motion for such a system with viscous damping is

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = f(t)$$

In the Rayleigh damping model the damping coefficient c can be expressed in terms of the mass m and the stiffness k as

$$c = \alpha_{dM} m + \beta_{dK} k$$

The Rayleigh damping proportional to mass and stiffness is added to the static weak term.

A complication with the Rayleigh damping model is to obtain good values for the damping parameters. A more physical damping measure is the relative damping, the ratio between actual and critical damping, often expressed as a percentage of the critical damping. Commonly used values of relative damping can be found in the literature.

It is possible to transform relative damping to Rayleigh damping parameters. The relative damping, ξ, for a specified pair of Rayleigh parameters, α_{dM} and β_{dK}, at a frequency, f, is

$$\xi = \frac{1}{2}\left(\frac{\alpha_{dM}}{2\pi f} + \beta_{dK} 2\pi f\right)$$

Using this relationship at two frequencies, f_1 and f_2, with different relative damping, ξ_1 and ξ_2, results in an equation system that can be solved for α_{dM} and β_{dK}:

$$\begin{bmatrix} \frac{1}{4\pi f_1} & \pi f_1 \\ \frac{1}{4\pi f_2} & \pi f_2 \end{bmatrix} \begin{bmatrix} \alpha_{dM} \\ \beta_{dK} \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

Using the same relative damping, $\xi_1 = \xi_2$, does not result in a constant damping factor inside the interval $f_1 < f < f_2$. It can be shown that the damping factor is lower inside the interval, as Figure 15-11 shows.

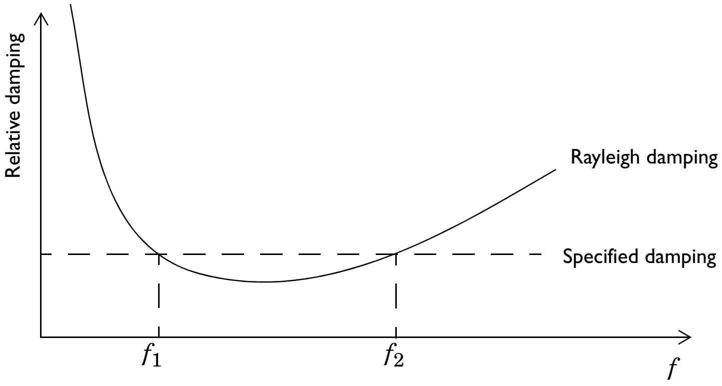


Figure 15-11: An example of Rayleigh damping.

16

Equation-Based Modeling

This chapter describes the use of the interfaces for mathematics and equation-based modeling, found under the **Mathematics** branch (Δu) when adding interfaces. With those interfaces you can solve various types of PDEs using different formulations. You can also solve ODEs and other global equations as well as create curvilinear coordinates.

The Mathematics Interfaces

The mathematics interfaces are a collection of tools for equation-based modeling and for performing special tasks, rather than for modeling specific physics. These interfaces support several PDE formulations as well as general ways to add ODEs, algebraic equations, other global (space-independent) equations, and curvilinear coordinates.

For a list of the available interfaces found under **Mathematics** branch (Δu) when adding interfaces, including the identifier names, see Physics Guide.

PDE INTERFACES

The PDE Interfaces branch contains Partial Differential Equation (PDE) interfaces for PDEs in coefficient form and general form, and for weak form PDEs on different geometry levels.

These interfaces are for entering PDEs in different forms:

- *Coefficient form* for PDEs conforming to the template explained in The Coefficient Form PDE Interfaces.
- *General form* for conservation laws and PDEs resulting from nonlinear material models. See The General Form PDE Interfaces.
- *Weak form* using the weak formulation of the PDE for maximum flexibility. See The Weak Form PDE Interfaces.
- The Wave Form PDE Interface solves PDEs with first-order derivatives in time and space using optimized algorithms with respect to speed and memory consumption.

Except for the Wave Form PDE, the PDE interfaces are available in domains, boundaries, edges, and at points. The interfaces for the different equation forms are identical except for the default node on the top geometric entity level. Also see Modeling with PDEs.

CLASSICAL PDES

The Classical PDE Interfaces branch contains some classical PDEs that are special cases of the Coefficient Form PDE: Laplace Equation, Poisson's Equation, Wave Equation, Heat Equation, Helmholtz Equation, and Convection-Diffusion Equation interfaces.

Also see Compact and Standard Notations for Classical PDEs.

ODE AND DAE INTERFACES

The ODE and DAE Interfaces are used to add global, space-independent equations that can represent additional states. The equations can be ODEs, algebraic equations, DAEs, and transcendental equations, either as global equations or as distributed ODEs/DAEs (on domains, boundaries, edges, or at points). For more information about global equations and ODEs, see Modeling with ODEs and DAEs.

EVENTS INTERFACE

The Events Interface is used to create solver events. An event can be explicit or implicit, and the difference is that for explicit events you must specify the exact time when the event occurs. When an event occurs, the solver stops and provides a possibility to reinitialize the values of states and dependent variables.

WALL DISTANCE INTERFACE

The Wall Distance Interface solves a modified eikonal equation for computing the distance to walls, which is an important quantity for turbulence modeling in fluid-flow simulations.

CURVILINEAR COORDINATES INTERFACE

Use the Curvilinear Coordinates interface to create a curvilinear coordinate system for defining anisotropic material properties following the shape of a geometric object. Three different methods are available for computing the coordinate system: a diffusion method, an elasticity method, and a flow method. You can also provide user-defined coordinate directions.

Modeling with PDEs

The physics interfaces in COMSOL Multiphysics and add on Modules use *partial differential equations*, PDEs, as a mathematical model of physical reality. You can access these PDEs in the following ways:

- The PDE Interfaces allow you to specify whole or part of your problem completely using PDEs. This approach may, for example, be suitable for modeling unusual equations from various fields of physics, or for learning mathematical modeling.
- Auxiliary equation-based nodes are available in all physics interfaces. These let you add extra equation contributions and constraints to the predefined mathematical model.
- Equation View nodes display the PDEs (on weak form) and constraints underlying the physics interfaces, and allow you to modify them.

This section describes the theory behind the PDE interfaces, but also contains information useful for understanding and modifying the mathematical models implemented in the physics interfaces.

- The PDE Interfaces
- The Wave Form PDE Interface
- About Auxiliary Equation-Based Nodes

About Equation Forms

Partial differential equations may be entered into COMSOL Multiphysics on four different formats:

- The General Form PDE
- The Coefficient Form PDE
- The Weak Form PDE
- The Wave Form PDE Interface

Which one to choose is mostly a matter of convenience: certain equations are quicker and easier to specify in one particular form. Internally, equations written on general or coefficient form are converted to weak form, which is therefore the most fundamental one. In particular, the weak form is closely linked to the theory behind the *finite element method*, FEM. In a similar way, the wave form is linked to a particular *discontinuous Galerkin* version of FEM, particularly suited for solving wave-propagation problems.

Notational Conventions

The PDE interfaces use a slightly different notation, compared to other physics interfaces and the associated documentation. The difference lies in the definition of the symbol ∇, pronounced *nabla* or *del*. For the physics interface equation sections and nodes (see Advanced Physics Sections), the ∇ symbol applied to a scalar or vector variable denotes the following coordinate system-independent *gradient*, *divergence*, and *curl* operations:

$$\nabla u = \mathrm{grad}(\mathrm{u})$$
$$\nabla \cdot \mathbf{u} = \mathrm{div}(\mathbf{u})$$
$$\nabla \times \mathbf{u} = \mathrm{curl}(\mathbf{u})$$

In the PDE interfaces, *nabla/del* is interpreted as the vector of partial derivatives:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right)$$

The spatial coordinates are denoted $x_1, \ldots, x_n$, where n represents the number of space dimensions. When applied to a scalar or vector in a Cartesian coordinate system, this definition leads to an expression that is identical in form to the gradient, divergence, or curl in the same coordinate system. The same does not apply, however, in curvilinear systems such as the one implied in an axisymmetric geometry.

For example, the divergence of a vector $\mathbf{u} = [u_r, u_z]$ in an axisymmetric cylindrical system is

$$\mathrm{div}(\mathbf{u}) = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} = \frac{1}{r}\frac{\partial}{\partial r}(ru_r) + \frac{\partial u_z}{\partial z}$$

while the PDE interface interpretation of *nabla/del* is:

$$\nabla \cdot \mathbf{u} = \frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z}$$

In practice, this means that to correctly implement equations containing the gradient, divergence, or curl in an axisymmetric geometry, you must compensate for the missing terms related to the curvature of the coordinate system. In particular, note that you must typically multiply the entire equation, as well as its boundary conditions, by a volume factor—in an axisymmetric geometry, for example, with a factor r—in order to recast it into one of COMSOL Multiphysics' equation forms.

The following related examples follow the same principle:

- The symbol Δ is the Laplace operator

$$\frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$$

- $\nabla \cdot (c\nabla u)$ means

$$\frac{\partial}{\partial x_1}\left(c\frac{\partial u}{\partial x_1}\right) + \ldots + \frac{\partial}{\partial x_n}\left(c\frac{\partial u}{\partial x_n}\right)$$

- $\beta \cdot \nabla u$ means

$$\beta_1\frac{\partial u}{\partial x_1} + \ldots + \beta_n\frac{\partial u}{\partial x_n}$$

where $\beta_1, \ldots, \beta_n$ are the components of the vector β.

The axisymmetric versions of physics interfaces take the cylindrical system into account, and no compensation is therefore needed.

- The PDE Interfaces
- The Wave Form PDE Interface

PDE Interface Variables

The following list shows symbolic expressions for quantities appearing in the definition of PDEs and corresponding variable names which can be used in PDE coefficients and are available for results evaluation and visualization

EXPRESSION	NAME	DESCRIPTION
u_i	`ui`	The solution variable (dependent variable)
$\frac{\partial u_i}{\partial x_i}$	`uixj`	The derivative of the solution variable u_i with respect to the spatial coordinate x_j, for example, `uy`
$(\nabla_T u_i)_j$	`uiTxj`	The x_j component of the gradient of u_i projected onto a boundary or edge, for example, `uTy`
$\frac{\partial^2 u_i}{\partial x_j \partial x_k}$	`uixjxk`	The second derivative of the solution variable u_i with respect to the spatial coordinates x_j and x_k, for example, `uxx`, `uxy`
$\frac{\partial u_i}{\partial t}$	`uit`	The derivative of the solution variable u_i with respect to time
$\frac{\partial^2 u_i}{\partial t^2}$	`uitt`	The second derivative of the solution variable u_i with respect to time
$\frac{\partial^2 u_i}{\partial x_j \partial t}$	`uixjt`	The mixed derivative of the solution variable u_i with respect to time and the spatial coordinate x_j

The General Form PDE

The General Form PDE interface provides a general interface for specifying and solving PDEs in the general form. The format is closely related to the conservation laws which govern many areas of physics. Assuming that you are working with a single dependent variable u, the general form reads:

$$\begin{cases} e_a\frac{\partial^2 u}{\partial t^2} + d_a\frac{\partial u}{\partial t} + \nabla\cdot\Gamma = f & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = g - qu + h^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ u = r & \text{on } \partial\Omega_d \end{cases} \qquad (16\text{-}1)$$

where

- Ω is the computational domain; the union of all domains
- $\partial\Omega$ is the domain boundary
- $\mathbf{n}$ is the outward unit normal vector on $\partial\Omega$

The first line (equation) of Equation 16-1is the PDE, which must be satisfied in Ω. The second, third and fourth equations are the boundary conditions, which must hold on $\partial\Omega$. The second equation is a generalization of a *Neumann* boundary condition. The third equation is a general *constraint*, of which the *Dirichlet* boundary condition on the fourth line is a special case.

The terms Γ, f, g, q, R and r are user-defined coefficients. They can be functions of the spatial coordinates, the solution u, and the space derivatives of u (see PDE Interface Variables), as well as of other predefined and user-defined variables. The coefficients f, g, q, R and r are scalar, whereas Γ is the *flux vector*.

In practical applications, Γ typically represents the flux of a conserved quantity such as heat, charge, mass or momentum. This flux is usually related in some empirical way, via a material law, to the gradient of the dependent variable. Therefore, Γ is usually a vector whose components are functions of derivatives of the dependent variable. The flux vector may also contain terms which are proportional to a velocity field when there is convective transport of the conserved quantity present. The structure of Equation 16-1 implies that the normal component of Γ is continuous across any surface in the interior of the domain, Ω.

BOUNDARY CONDITIONS FOR THE GENERAL FORM PDE

In finite element terminology, the boundary condition on the second line of Equation 16-2, corresponding to a Neumann boundary condition, is called a *natural boundary condition*, because it does not occur explicitly in the weak form of the PDE problem. In the PDE interfaces, the corresponding condition is called a *flux* or *source*, because it specifies the value of the numerical flux Γ at the boundary.

Constraints and Dirichlet conditions are also known as *essential boundary conditions* in finite element theory, because they impose a restriction on the trial space which is not part of the main equation. In the PDE interfaces, a distinction is made between Dirichlet boundary conditions and *constraints*. The general constraint on line 3 of Equation 16-2 specifies that an arbitrary expression is equal to zero on the boundary, $R = 0$. The Dirichlet condition on line 4 of the same equation is a special case directly specifying the value of the dependent variable at the boundary, $u = r$. This makes the constraint a more general boundary condition.

The term $-h^{\mathrm{T}}\mu$ in the generalized Neumann condition is a reaction term enforcing the constraint $R = 0$. When reaction terms are applied symmetrically on all dependent variables,

$$h = \frac{dR}{du}$$

but also other definitions are possible. The variable μ is a Lagrange multiplier, which is eliminated by the solvers when using standard constraints, and therefore does not normally appear explicitly in equations.

For details about the time-dependent and eigenvalue formulations, see Solving Time-Dependent Problems and Solving Eigenvalue Problems.

- The General Form PDE Interfaces
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs

The Coefficient Form PDE

The Coefficient Form PDE provides a general interface for specifying and solving many well-known PDEs in the coefficient form.

Many PDEs originating from physics and other fields may be cast into a generic form containing derivatives up to second order in both time and space, but no mixed derivatives. In COMSOL multiphysics, you can define a PDE of this type by specifying coefficients for the derivatives of different orders. This results in a *coefficient form* PDE, which for one dependent variable u reads:

$$
\begin{cases}
e_a\dfrac{\partial^2 u}{\partial t^2}+d_a\dfrac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au=f & \text{in }\Omega\\
\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)=g-qu+h^T\mu & \text{on }\partial\Omega\\
0=R & \text{on }\partial\Omega_c\\
u=r & \text{on }\partial\Omega_d
\end{cases}
\tag{16-2}
$$

where

- Ω is the computational domain; the union of all domains
- $\partial\Omega$ is the domain boundary
- $\mathbf{n}$ is the outward unit normal vector on $\partial\Omega$

The first line (equation) of Equation 16-2is the PDE, which must be satisfied in Ω. The second and third equations are the boundary conditions, which must hold on $\partial\Omega$. The second equation is a generalization of a Neumann boundary condition. The third equation is a general constraint, with a Dirichlet boundary condition as a special case. For more information about the boundary conditions, see The General Form PDE.

To define a PDE on coefficient form in one of the PDE interfaces, you specify the coefficients c, α, γ, β, and a and the boundary terms f, g, R and r. They can all be functions of the spatial coordinates as well as of dependent variables and other predefined or user-defined variables and parameters. A PDE is guaranteed to be *linear* when the coefficients vary only with the spatial coordinates (or are constants). A PDE is *nonlinear* if the c, α, β, a, h or q coefficients depend on u or its derivatives (for example, the components of ∇u), or if γ, f, g, R or r are nonlinear in u.

For a single dependent variable u, all the coefficients in the above equation are scalars except α, β, and γ, which are vectors with n components. The coefficient c may be given alternatively as a scalar or an n-by-n matrix to model anisotropic materials. When the coefficient form is used for modeling a system of equations, the coefficients are extended with additional vector and matrix dimensions referring to the dependent variable index. See further Multiple Dependent Variables—Equation Systems.

COEFFICIENT FORM VERSUS GENERAL FORM

Comparing Equation 16-2 to Equation 16-1 it is easy to see that the coefficient form is just a special case of the general form. In fact, applying the following substitutions in the general form, Equation 16-1, turns it into the coefficient form:

$$
\begin{aligned}
\Gamma &= -c\nabla u-\alpha u+\gamma\\
F &= f-\beta\nabla u-au
\end{aligned}
\tag{16-3}
$$

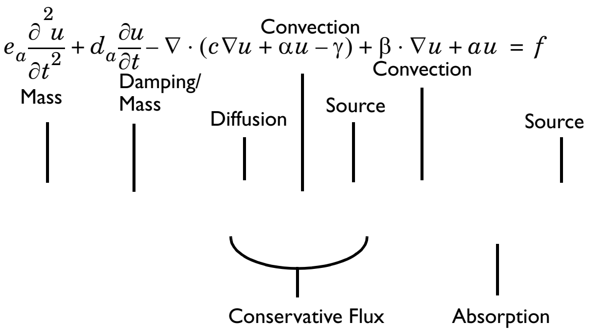
This duality lets you choose the representation in which it is easiest to implement a particular PDE. There is no difference in performance.

INTERPRETING PDE COEFFICIENTS

The COMSOL Multiphysics PDE formulations can model a variety of problems, but this guide as well as the interface uses descriptive names for the coefficients that fall within the realm of continuum mechanics and mass transfer. For the coefficient form PDE:

- e_{a} is the mass coefficient.
- d_{a} is a damping coefficient or a mass coefficient.

- c is the diffusion coefficient.
- α is the conservative flux convection coefficient.
- β is the convection coefficient.
- a is the absorption coefficient.
- γ is the conservative flux source term.
- f is the source term.



$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u+\alpha u-\gamma)+\beta\cdot\nabla u+au=f$$

For the Neumann boundary condition of the coefficient form

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)=g-qu+h^T\mu$$

- g is the boundary source term.
- q is the boundary absorption coefficient.

> There are many interesting PDE problems to which these interpretations do not apply. For example, a time-harmonic PDE such as the Helmholtz equation represents a time-dependent phenomenon transformed into the frequency domain, making the a coefficient a mass rather than absorption term.

COMPACT AND STANDARD NOTATIONS FOR CLASSICAL PDES

Many classical PDEs are instances of the coefficient form PDE. The classical PDEs have their own interfaces which are found under the **Mathematics>Classical PDEs** branch (∇^2) when adding an interface. Table 16-1 shows the available classical PDEs using two notations: the compact notation of vector analysis (used in this documentation) and an expanded component notation.

TABLE 16-1: CLASSICAL PDES IN COMPACT AND STANDARD NOTATION

EQUATION	COMPACT NOTATION	STANDARD NOTATION (2D)
Laplace's equation	$-\nabla\cdot(\nabla u)=0$	$-\frac{\partial}{\partial x}\frac{\partial u}{\partial x}-\frac{\partial}{\partial y}\frac{\partial u}{\partial y}=0$
Poisson's equation	$-\nabla\cdot(c\nabla u)=f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)=f$
Helmholtz equation	$-\nabla\cdot(c\nabla u)+au=f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au=f$
Heat equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)=f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)=f$

TABLE 16-1: CLASSICAL PDES IN COMPACT AND STANDARD NOTATION

EQUATION	COMPACT NOTATION	STANDARD NOTATION (2D)
Wave equation	$e_a\frac{\partial^2 u}{\partial t^2} - \nabla\cdot(c\nabla u) = f$	$e_a\frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Convection-diffusion equation	$d_a\frac{\partial u}{\partial t} - \nabla\cdot(c\nabla u) + \beta\cdot\nabla u = f$	$d_a\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)$ $+ \beta_x\frac{\partial u}{\partial x} + \beta_y\frac{\partial u}{\partial y} = f$

The default values are 1 for f and c and -1 for a, so the default Helmholtz equation, for example, is $-\Delta u - u = 1$.

- Multiple Dependent Variables—Equation Systems
- The Coefficient Form PDE Interfaces
- The PDE Interfaces
- The Classical PDE Interfaces
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs
- Modeling Anisotropic Materials

Multiple Dependent Variables—Equation Systems

All PDE interfaces and equation forms support multiple dependent variables. In the case of several dependent variables u_1, u_2, …, u_N, a general form system of equations takes the following form:

$$\begin{cases} e_a^{lk}\frac{\partial^2 u_k}{\partial t^2} + d_a^{lk}\frac{\partial u_k}{\partial t} + \nabla\cdot\Gamma_l = F_l & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma_l = G_l + h_{ml}\mu_m & \text{on } \partial\Omega \\ 0 = R_m & \text{on } \partial\Omega_c \\ u_n = r_n & \text{on } \partial\Omega_d \end{cases} \qquad (16\text{-}4)$$

The equation index l and k ranges from 1 to N, while the general constraint index m ranges from 1 to M_c and the Dirichlet condition index n ranges from 1 to M_d. The total number of constraints is therefore $M = M_c + M_d$. This discussion uses the summation convention. F_l, G_l, R_m and r_n are scalars, whereas Γ_l is a spatial vector. The mass and damping coefficients e_a and d_a are N-by-N matrices, while the constraint force Jacobian h is an M-by-N matrix.Note that in this case there are several Lagrange multipliers: μ_1, μ_2,…, μ_M.

For a more compact form, let $\mathbf{u}$ be a vector with components u_k, let Γ be a matrix with components Γ_{lj}, and so on. Then the system of equations takes on the same form as given in Equation 16-1 for a single dependent variable.

It is also possible to write the system entirely on component form, where Γ_{lj} are components of the vector Γ_l, and n_j components of the normal vector $\mathbf{n}$. Then the system of equations becomes:

$$\begin{cases} e_a^{lk}\dfrac{\partial^2 u_k}{\partial t^2} + d_a^{lk}\dfrac{\partial u_k}{\partial t} + \dfrac{\partial}{\partial x_j}(\Gamma_{lj}) = F_l & \text{in } \Omega \\ -n_j\Gamma_{lj} = G_l + h_{ml}\mu_m & \text{on } \partial\Omega \\ 0 = R_m & \text{on } \partial\Omega_c \\ u_n = r_n & \text{on } \partial\Omega_d \end{cases}$$

- The General Form PDE
- The General Form PDE Interfaces
- Boundary Condition Types

THE COEFFICIENT FORM EQUATION SYSTEM

The coefficient form of an equation system with N dependent variables $u_1, u_2, \ldots, u_N$ can be easily obtained from the general form PDE shown in Equation using the substitutions:

$$\begin{cases} \Gamma_{lj} = -c^{lkji}\dfrac{\partial u_k}{\partial x_i} - \alpha^{lkj}u_k + \gamma^{lj} \\ F_l = f_l - \beta^{lki}\dfrac{\partial u_k}{\partial x_i} - a^{lk}u_k \end{cases}$$

Where index k and l run over dependent variables form 1 to N, while index i and j run over space dimensions from 1 to K. This means that for the case of a system of equations with N dependent variables in K space dimensions, the coefficients have the following sizes:

- e_a is an N-by-N matrix
- d_a is an N-by-N matrix
- c is an N-by-N-by-K-by-K four-dimensional array
- α is an N-by-N-by-K three-dimensional array
- β is an N-by-N-by-K three-dimensional array
- a is an N-by-N matrix
- f is an N-vector
- g is an N-vector
- q is an N-by-N matrix

- The Coefficient Form PDE
- The Coefficient Form PDE Interfaces
- Boundary Condition Types

Solving Time-Dependent Problems

The general form equation shown in Equation 16-1 as well as the coefficient form equation in Equation 16-2 contain time-derivative terms of the same form. These terms only take effect for Time Dependent, Eigenvalue and

Eigenfrequency study steps, and derived versions of these. When solving a Stationary, Frequency Domain or similar study step, all time derivatives are assumed to be zero. In this case, the value of the e_a and d_a coefficients does not matter.

To activate the d_a and e_a coefficients and convert the model into a time-dependent model, select a Time Dependent study.

When solving a Time Dependent study step, the *mass coefficient*, e_a, becomes important. The name mass coefficient, or *mass matrix* in case of a system of equations, stems from the fact that in many physics applications, e_a contains the mass density. The d_a coefficient in such equations usually represents damping of wave-like phenomena. However, if $e_a = 0$, then d_a is often called the mass coefficient instead. The default settings are $e_a = 0$ and $d_a = 1$, representing a parabolic time-dependent PDE such as the heat equation. Using $e_a = 1$ and $d_a = 0$ represents an undamped wave equation.

When solving a Time Dependent study step, the time variable is called `t` and can be used anywhere in equation coefficients. For other study steps, `t` is undefined. If you want to solve a model that depends explicitly on time using a Stationary study, you must first define a model parameter called `t` and give it a suitable value.

If, for a system of equations, the e_a matrix is nonzero and singular, or if $e_a = 0$ and d_a is singular, the system becomes a *differential-algebraic equation* (DAE) systems. The COMSOL solvers for time-dependent problems handle DAEs.

Time-Dependent Solver

USING MIXED SPACE-TIME DERIVATIVES

The coefficient forms in equation Equation 16-2 only contains coefficients for pure space and time derivatives up to second order. The only directly available time-derivative coefficients are therefore e_a and d_a, using the subscript a because they are similar to the a coefficient in the absorption term except that they multiply $\partial^2 u/\partial t^2$ and $\partial u/\partial t$ instead of u. In analogy, it is possible to define coefficients e_c, $\mathbf{e}_\alpha$, $\mathbf{e}_\beta$ and d_c, $\mathbf{d}_\alpha$, $\mathbf{d}_\beta$ for mixed space-time derivatives, such that the equation becomes instead

$$e_a\frac{\partial^2 u}{\partial t^2} + \nabla\cdot\left(-e_c\nabla\frac{d^2 u}{dt^2} - \mathbf{e}_\alpha\frac{d^2 u}{dt^2}\right) + \mathbf{e}_\beta\cdot\nabla\frac{d^2 u}{dt^2} +$$

$$d_a\frac{\partial u}{\partial t} + \nabla\cdot\left(-d_c\nabla\frac{\partial u}{\partial t} - \mathbf{d}_\alpha\frac{\partial u}{\partial t}\right) + \mathbf{d}_\beta\cdot\nabla\frac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u + \ldots) = \ldots$$

These mixed coefficients are not directly available in the general or coefficient form PDE models. Instead, enter them in the existing γ and f terms:

In 1D, add `-d_c*uxt-d_al*ut` to the γ term, and add `-d_be*uxt` to the f term, and similarly for second-order derivatives.

In 2D, add `-d_c*uxt-d_al1*ut` to the first γ component, and add `-d_c*uyt-d_al2*ut` to the second γ component. Add `-d_be1*uxt-d_be2*uyt` to the f term, and similarly for second-order derivatives.

USING TIME DERIVATIVES IN BOUNDARY CONDITIONS

To specify a flux or source boundary condition containing time-derivative terms as in

$$\mathbf{n}\cdot(c\nabla u+\ldots) = g-e_q\frac{\partial^2 u}{\partial t^2}-d_q\frac{\partial u}{\partial t}-qu+h^T\mu\,,$$

simply add the terms `-e_q*utt-d_q*ut` to the g term, and provide appropriate values or expressions for the coefficients e_q and d_q in, for example, a Global Equations settings window.

Constraints and Dirichlet boundary conditions must not contain time derivatives like `ut` and `utt` in the R and r coefficients unless they are enforced weakly, using weak constraints. See Boundary Conditions.

Solving Eigenvalue Problems

THE EIGENVALUE PDE

When solving a PDE using an Eigenvalue study step, COMSOL assumes that all dependent variables vary with time as $u(t)=\hat{u}e^{-\lambda t}$, where $\hat{u}$ is a complex amplitude field. Therefore the time derivatives in Equation 16-1 and Equation 16-2 are interpreted as

$$\frac{\partial u}{\partial t} = -\lambda\hat{u}$$

$$\frac{\partial^2 u}{\partial t^2} = \lambda^2\hat{u}$$

which for example leads to the general form eigenvalue PDE

$$\begin{cases}\lambda^2\hat{u}-\lambda\hat{u}+\nabla\cdot\Gamma=f & \text{in}\Omega\\ -\mathbf{n}\cdot\Gamma=g-q\hat{u}+h^T\hat{\mu} & \text{on }\partial\Omega\\ 0=R & \text{on }\partial\Omega_c\\ \hat{u}=r & \text{on }\partial\Omega_d\end{cases}$$

The eigenvalue solver further ignores any source or flux terms which are independent of the dependent variables.

BOUNDARY CONDITIONS IN EIGENVALUE PROBLEMS

Boundary conditions are treated as homogeneous for eigenvalue and eigenfrequency studies. It means, for example, that when using a Dirichlet boundary condition such as $u=7$, it is treated as $u=0$ when you use eigenvalue or eigenfrequency study steps. For nonlinear problems the eigenvalue solver is linearizing the problem, including the constraints, around a linearization point for the dependent variables and a eigenvalue linearization point. For a nonlinear constraint (for u),

$$f(u) = 0$$

the constraint

$$f_u(u_0) \cdot u = 0$$

is used when you run eigenvalue or eigenfrequency study steps. The eigenvalue itself is not supported in constraints.

THE EIGENVALUES AND THE LAMBDA VARIABLE

As an alternative to defining eigenvalue PDEs using the time-derivative coefficients e_a and d_a, you can write the eigenvalue explicitly in the equations using the variable name `lambda`. For example, instead of specifying $e_a = 1$ you can set a = `lamda^2` with exactly the same result. In many cases, this formulation is preferable, in particular when the eigenvalue problem does not arise from a time derivative in a time-harmonic assumption.

After solving an eigenvalue problem, the eigenvalue is always available for postprocessing under the variable name `lambda`, independently of whether the problem has been specified using the e_a and d_a coefficients or using the variable `lambda.`

Eigenfrequency studies are exactly analogous to Eigenvalue studies except that they in addition define the variable `freq` using the definition `freq` = $i\lambda/(2\pi)$. The variable name `freq` may be used in equations and postprocessing in the same way as `lambda`.

The eigenvalue solvers solve eigenvalue problems which are at most quadratic polynomials in the eigenvalue `lambda` exactly in one step. Therefore, damped eigenvalue solutions are easily found when both e_a and d_a are nonzero. Using the variable `lambda`, more complicated eigenvalue problems can be specified. Such problems must be solved using an iterative procedure.

Each time you run the eigenvalue solver, the PDE is expanded in a Taylor series in `lambda` around the eigenvalue linearization point λ_0. Only the linear and quadratic terms are retained, while higher order terms are dropped. Running the solver repeatedly, updating the eigenvalue linearization point to the last eigenvalue found, usually converges to an eigenvalue solving the full nonlinear eigenvalue problem.

Eigenvalue Solver and Eigenvalue.

About Weak Form Modeling

Do not be misled by the term “weak;” the weak form is very powerful and flexible. The term *weak form* is borrowed from mathematics, but in this context it has a slightly different meaning; this implementation incorporates capabilities in addition to those defined in the mathematical weak form. Moreover, knowledge of the mathematical weak form is not a prerequisite to using the COMSOL Multiphysics implementation.

The distinguishing characteristics of the weak form in COMSOL Multiphysics are that it makes it possible to:

- Enter certain equations that can be derived from an energy principle in a very compact and convenient form. Such equations, for example, arise in structural mechanics.
- Add and modify nonstandard constraints, such as various contact and friction models.

- Build models with extra equations on boundaries, edges, and points.
- Use the *test operator* to conveniently work with problems in variational calculus and parametric optimization. For more information about the test operator and other operators, see Operators, Functions, and Constants.

All physics interfaces are implemented as weak form equations which you may study and modify in the Equation View nodes. COMSOL Multiphysics also converts all equation-based models specified in The Coefficient Form PDE Interfaces and The General Form PDE Interfaces to the weak form before solving.

In addition, it is possible in COMSOL Multiphysics to add extra weak form contributions and auxiliary variables to any physics interface in the model.

- Physics Nodes—Equation Section and Equation View
- About Auxiliary Equation-Based Nodes
- The Weak Form PDE Interfaces and Weak Form PDE

Introduction to the Weak Form

The general form and coefficient form PDEs in equations Equation 16-1 and Equation 16-2 specify PDEs on *strong form*, in the sense that they in principle require the PDE to be satisfied at every point. And for this to be possible, all terms must be sufficiently continuous for derivatives and well-defined pointwise values to exist. In many cases, the natural phenomena a PDE intends to model are in fact best described as discontinuous, and may also contain source terms which are only defined as a total over a small region, without a well-defined pointwise value.

In these situations a weak equation turns out to be a better model of physics than can be provided by the more commonly used strong form PDEs. In addition, the weak form is particularly suitable for discretization and numerical solution using the finite element method. One reason for this is precisely the lower continuity requirement on the solution, which only needs to be sufficiently smooth on each mesh element separately.

EXAMPLE: CONVERSION FROM GENERAL FORM TO WEAK FORM

As an example, consider the general form presented in Equation 16-1, in particular the stationary form of the domain equation:

$$\nabla \cdot \Gamma = f$$

Assuming a single dependent variable u, introduce a corresponding arbitrary test function υ. Multiply the equation by this test function and integrate over the domain:

$$\int_{\Omega} \upsilon \nabla \cdot \Gamma dV = \int_{\Omega} \upsilon f dV \qquad (16\text{-}5)$$

Requiring this integral equation to hold is clearly a weaker statement than the original equation, in particular if Equation 16-5 is required to hold only for all test functions υ from a limited class of functions. In the finite element method, the test functions υ (and also solution u) are usually limited to the set of piecewise polynomials of a given order on each mesh element.

This polynomial can also be written as a sum of individual *shape functions*. Therefore, the original strong form PDE is transformed into a weak form equation which must only be satisfied in a local integral sense over each shape function. When you increase the number of shape functions by refining the mesh or increasing the polynomial order, you simultaneously decrease the space of solutions u which can possibly satisfy Equation 16-5. Therefore, well-posed and consistent finite element formulations converge towards the single solution u which satisfies the original strong form PDE.

To further simplify solution of Equation 16-5, the left-hand side integral can be integrated by parts, using Gauss law:

$$-\int_{\Omega}\nabla\upsilon\cdot\Gamma dV+\int_{\delta\Omega}\upsilon\mathbf{n}\cdot\Gamma dA=\int_{\Omega}\upsilon f dV \tag{16-6}$$

This has two main advantages. First of all, it reduces the maximum order of spatial derivatives. If Γ is a function of the gradient of u, for example $\Gamma = -c\nabla u - \alpha u + \gamma$ as in the coefficient form PDE, the transformed weak equation now contains only first-order derivatives compared to second-order derivatives in the original strong form PDE. Secondly, it makes it clear what the ***natural boundary condition*** is for this equation. The second integral on the left-hand side disappears if the normal component of Γ vanishes on the boundary. Alternately, if the value of the normal component is known, for example such that

$$-\mathbf{n}\cdot\Gamma = g - qu + h^T\mu \tag{16-7}$$

on $\delta\Omega$, this value can be inserted as a boundary condition into the weak form equation, which then becomes

$$-\int_{\Omega}\nabla\upsilon\cdot\Gamma dV=\int_{\Omega}\upsilon f dV+\int_{\delta\Omega}\upsilon G dA \tag{16-8}$$

This final weak formulation of the standard general form PDE therefore also explains why the Neumann boundary condition on the second line of Equation 16-1 looks the way it does.

The Weak Form PDE

The Weak Form PDE provides a general interface for specifying and solving PDEs in the weak form.

The weak form does not define any coefficients and does not even separate the different equations in a system of equations. When specifying a PDE and its boundary conditions on weak form, you specify contributions to a generic weak equation which for a 3D model reads:

$$0=\sum_{i=1}^{N_3}\int_{\Omega_i}W_3^i dV_i+\sum_{j=1}^{N_2}\int_{\delta\Omega_j}W_2^j dA_j+\sum_{k=1}^{N_1}\int_{\delta^2\Omega_k}W_1^k dL_k+\sum_{m=1}^{N_0}\sum_{\delta^3\Omega_m}W_0^m \tag{16-9}$$

This weak equation has

- N_3 domain contributions $W^i{}_3$, each integrated over domain selection Ω_i
- N_2 boundary contributions $W^j{}_2$, each integrated over boundary selection $\delta\Omega_j$
- N_1 edge contributions $W^k{}_1$, each integrated over edge selection $\delta^2\Omega_k$
- N_0 point contributions $W^m{}_0$, each summed over point selection $\delta^3\Omega_m$

Note that all contributions are summed into the same integral equation without any particular order. Therefore, whether you write two equations in separate contributions or sum them into one single contribution normally does not matter. There is, however, a small caveat: the index on dV_i, dA_j, and dL_k indicate that each contribution may be integrated in a different way. The integration can be performed with respect to either material or spatial coordinates and, in addition, using different numerical quadrature orders. But while working inside a single PDE interface, all contributions are integrated in the same way.

USING THE TEST OPERATOR

When specifying a weak contribution, you may use all variables normally available for evaluation in equation contributions and during postprocessing. This includes independent variables (coordinates), dependent variables and their derivatives, and other predefined and user-defined variables, parameters and constants. In addition, you must use the `test` operator to distinguish between test functions and the solution.

> The `test` operator must always occur linearly in each weak form contribution. Contributions or terms without any `test` operator are ignored, while terms nonlinear in the test operator are considered an error.

In many cases, it is sufficient to let the test operator act directly on the dependent variables and their derivatives. For example, the weak form of the heat equation in two dimensions is `q*test(u)-k*(ux*test(ux)+uy*test(uy))`, where `q` is a heat source and `k` is the thermal conductivity. In other cases, it is more convenient to insert an expression or user-defined variable into the test function. One example of this is geometrically nonlinear solid mechanics, where the weak form can be written as a sum of terms `S_i*test(E_i)` where `S_i` is a stress measure and `E_i` its conjugate strain measure.

When a nonlinear expression, like for example a Green-Lagrange strain, is inserted into the `test` operator, the operator acts as a linear differential operator. This means that the argument expression is effectively first differentiated with respect to each dependent variable it contains and the results multiplied by the test function of the corresponding variable. Therefore, the heat equation may alternately be implemented as `q*test(u)-0.5*k*test(ux^2+uy^2)`. Using the chain rule on the second term with the `test` operator as differential operator, returns the standard weak form of the equation given above.

> For more information about the test operator and other operators, see Operators, Functions, and Constants.

Specifying and Interpreting Boundary Conditions

The formulation of the boundary conditions in general form (Equation 16-1) and coefficient form (Equation 16-2) imposes both Dirichlet and Neumann conditions at the same time:

$$\begin{cases} -\mathbf{n}\cdot\Gamma = g - qu + h^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ u = r & \text{on } \partial\Omega_d \end{cases}$$

where Γ is the *flux vector* ($\Gamma = -c\nabla u - \alpha u + \gamma$ for a coefficient form equation) and $\delta\Omega_c$ and $\delta\Omega_d$ are parts of the overall boundary, $\delta\Omega$, where general constraints and Dirichlet conditions have been specified. Combining conditions of different types on the same boundary is possible because of a new dependent variable μ, which is defined only on the boundary. This unknown variable μ is called a *Lagrange multiplier*, and usually has a physical interpretation. For example, in structural mechanics problems the Lagrange multiplier equals the reaction forces on the boundary.

The factor h^T in the Neumann boundary condition is the *constraint force Jacobian*. It decides how the Lagrange multipliers enforcing the constraint are scaled and distributed over the equations. The default settings in a Constraint node uses

$$h^T = -\left(\frac{dR}{du}\right)^T$$

while a Dirichlet Boundary Condition node by default corresponds to $h^T = 1$. For example:

- The Dirichlet condition is $u = r$ and the default constraint settings imply $h^T = 1$. The Neumann condition becomes:

  $$-\mathbf{n} \cdot \Gamma = g - qu + \mu$$

  The Lagrange multiplier, μ, adjusts so as to satisfy the requested Dirichlet condition. Specifying a nonzero g changes the value of the Lagrange multiplier on the same boundary but does not affect the actual solution u. Therefore, this equation can usually be ignored, leaving effectively a pure Dirichlet condition.
- When no constraint is applied on a boundary, the value of R is zero, or equivalently, the Dirichlet condition reads $0 = 0$. Therefore h^T is zero and the Neumann condition is:

  $$-\mathbf{n} \cdot \Gamma = g - qu$$

  This is the generalized Neumann condition without a Lagrange multiplier.

- Boundary Condition Types
- The PDE Interfaces

EXAMPLE: SYSTEM OF TWO VARIABLES IN THE GENERAL FORM

The following example demonstrates a number of possible boundary condition combinations for a stationary system with two dependent variables u_1 and u_2 and two constraints when reaction terms are applied symmetrically on all physics. This is the default, and most useful, implementation. Written in general form:

$$\begin{cases} \nabla \cdot \Gamma_1 = F_1 & \text{in } \Omega \\ \nabla \cdot \Gamma_2 = F_2 & \text{in } \Omega \end{cases}$$

with the default Neumann boundary conditions

$$\begin{cases} -\mathbf{n} \cdot \Gamma_1 = G_1 + \dfrac{\partial R_1}{\partial u_1}\mu_1 + \dfrac{\partial R_2}{\partial u_1}\mu_2 & \text{on } \partial\Omega \\ -\mathbf{n} \cdot \Gamma_2 = G_2 + \dfrac{\partial R_1}{\partial u_2}\mu_1 + \dfrac{\partial R_2}{\partial u_2}\mu_2 & \text{on } \partial\Omega \end{cases}$$

writing out the symmetric application of reaction terms on all dependent variables, and the Dirichlet boundary conditions:

$$\begin{cases} 0 = R_1 & \text{on } \partial\Omega \\ 0 = R_2 & \text{on } \partial\Omega \end{cases}$$

The same set of boundary conditions are accessible in all PDE interfaces. To illustrate the flexibility of the Constraint boundary condition $R = 0$, consider these cases:

Case 1: Let $R_1 = R_2 = 0$. Then the Dirichlet boundary conditions give $0 = 0$. In addition, the terms containing the Lagrange multipliers disappear from the Neumann boundary condition. Thus you have only the Neumann boundary conditions:

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

Case 2: Let $R_1 = r_1 - u_1$ and $R_2 = r_2 - u_2$. Then the Dirichlet conditions are the usual $u_1 = r_1$ and $u_2 = r_2$. Using default settings for the constraint reaction terms,

$$h = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and the Neumann boundary conditions become:

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 - \mu_2 & \text{on } \partial\Omega \end{cases}$$

These last equations impose no restrictions on u_1 or u_2, because the Lagrange multipliers μ_1 and μ_2 always adjust so as to fulfill the Dirichlet conditions. In this case, ignore the Neumann boundary conditions.

Case 3: Let $R_1 = r_1 - u_1$ and $R_2 = 0$. Then the Dirichlet conditions are

$$\begin{cases} 0 = r_1 - u_1 & \text{on } \partial\Omega \\ 0 = 0 & \text{on } \partial\Omega \end{cases}$$

and the default Neumann conditions including reaction terms are:

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

The first Neumann condition can be ignored because it imposes no restriction on u_1 or u_2. You effectively have only the Dirichlet condition on u_1 together with the second Neumann condition.

Case 4: The same as Case 3 but with the two PDEs interchanged (Γ_1 and Γ_2 as well as F_1 and F_2). Then the PDEs are:

$$\begin{cases} \nabla\cdot\Gamma_2 = F_2 & \text{in } \Omega \\ \nabla\cdot\Gamma_1 = F_1 & \text{in } \Omega \end{cases}$$

The Dirichlet condition is similar to that in Case 3: $u_1 = r_1$. By default, the Neumann conditions then become:

$$\begin{cases} -\mathbf{n}\cdot\Gamma_2 = G_2 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_1 = G_1 & \text{on } \partial\Omega \end{cases}$$

Effectively, you have only the Neumann condition $-\mathbf{n} \cdot \Gamma_1 = G_1$. In comparison with Case 3, the PDEs and the Dirichlet conditions are identical, while the Neumann conditions are different. In fact, both the Dirichlet and the Neumann conditions are now applied on u_1, and nothing is specified for u_2.

This example shows that when mixing Dirichlet and Neumann conditions on Coefficient Form PDEs and General Form PDEs, the ordering of the equations and the dependent variables are important. However, the ordering of the Dirichlet conditions does not matter because the different Lagrange multipliers are for all practical purposes indistinguishable from each other.

Case 5: Finally, let $R_1 = u_2 - u_1$ and $R_2 = 0$. Also, assume that u_1 and u_2 exist on two adjacent domains rather than on the same domain. The normal vectors as seen from the two sides are then $\mathbf{n}_1 = -\mathbf{n}_2 = \mathbf{n}$. Then the Dirichlet conditions are:

$$\begin{cases} 0 = u_2 - u_1 & \text{on } \partial\Omega \\ 0 = 0 & \text{on } \partial\Omega \end{cases}$$

and the Neumann conditions using the default symmetric reaction terms are:

$$\begin{cases} -\mathbf{n}_1 \cdot \Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}_2 \cdot \Gamma_2 = G_2 + \mu_1 & \text{on } \partial\Omega \end{cases}$$

The same Lagrange multiplier now appears in both Neumann conditions, which can have different definitions of Γ and G. Therefore, contrary to Cases 2 and 3, the Neumann conditions cannot be ignored. Instead, adding the two conditions, it becomes apparent that the solution and flux on the boundary must fulfill:

$$\begin{cases} 0 = u_2 - u_1 & \text{on } \partial\Omega \\ -\mathbf{n}_1 \cdot \Gamma_1 - \mathbf{n}_2 \cdot \Gamma_2 = G_1 + G_2 & \text{on } \partial\Omega \end{cases}$$

In particular, if $G_1 = G_2 = 0$, the last condition simplifies to:

$$-\mathbf{n} \cdot (\Gamma_1 - \Gamma_2) = 0 \text{ on } \partial\Omega$$

This means that both the variables u_1 and u_2 and the corresponding fluxes are equal at the boundary. In fact, if u_1 and u_2 represent the same quantity, this is the same continuity condition that holds implicitly at every mesh element boundary in the model, where nothing else has been specified.

In all of these examples, the values of the Lagrange multipliers do not matter. However, they often have a physical significance. In structural mechanics, the term $h^{\mathrm{T}}\mu$ in the Neumann condition is the reaction force necessary to satisfy the kinematic constraints described by the Dirichlet conditions.

Symmetric and Nonsymmetric Constraints

Constraints formulated through the coefficient R in The Coefficient Form PDE Interfaces and The General Form PDE Interfaces by default give rise to *globally symmetric bidirectional* constraints. This happens when the constraint settings specify that reaction terms are to be applied symmetrically on all physics.

A bidirectional symmetric constraint dictates exactly how the flux conditions (or Neumann boundary conditions) are influenced by the constraint force. For the coefficient form, the flux condition is in this case

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma) \;=\; g-qu-\left(\frac{\partial R}{\partial u}\right)^{T}\mu$$

and for the general form, the flux condition is

$$-\mathbf{n}\cdot\Gamma \;=\; g-qu-\left(\frac{\partial R}{\partial u}\right)^{T}\mu$$

The last term on the right-hand side in both expressions is the globally symmetric constraint reaction term, or *generalized constraint force*. Thus, with symmetric constraints a flux condition cannot be enforced independently of the constraints.

In mathematics, as well as in multiphysics modeling, it is often necessary to enforce Neumann conditions and Dirichlet conditions more freely than what is possible through symmetric constraints. As an example, consider the general form and assume that you want to enforce the boundary conditions:

$$\begin{cases} 0 = r_1-u_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

If $r_1=r_1(u_2)$, the first condition is fulfilled but not the second if the default reaction term definition is used. This is because the globally symmetric constraint force is not zero:

$$-\mathbf{n}\cdot\Gamma_2 \;=\; G_2+\frac{\partial R_1}{\partial u_2}\mu_1+\frac{\partial R_2}{\partial u_2}\mu_2 \;=\; G_2+\frac{\partial r_1}{\partial u_2}\mu_1\neq G_2$$

To remedy this limitation with bidirectional constraints, the **Constraint Settings** section allows you to **Apply reaction terms on** either dependent variables from **Current physics (internally symmetric)** or **Individual dependent variables**. Both options imply a unidirectional and possibly nonsymmetric constraint in the sense that some dependent variables are considered as constants for the purpose of enforcing the constraint.

To display the **Constraint Settings** section in Constraint nodes, click the **Show** button ( ) on the **Model Builder** tool bar and select **Advanced Physics Options**.

When constraint reaction terms are applied only on the current physics, flux conditions in other interfaces are left untouched by the constraint. If reaction terms are applied only to individual variables, this leaves flux conditions untouched on all but the specific variables. For the above example, both settings have the same desired effect if u_1 and u_2 belong to different interfaces. If these belong to the same interface, applying reaction terms to **Current physics (internally symmetric)** has the same effect as the default application to **All physics (symmetric)**.

In multiphysics modeling, unidirectional constraints are, for example, necessary for the following boundary conditions:

- Normal-direction constraints on a moving mesh, where the mesh motion is part of the problem. These conditions are of the type $\mathbf{n} \cdot \mathbf{u} - r = 0$ where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the boundary normal, $\mathbf{u}$ is a vector field (displacements or velocity), and $\mathbf{x}$ is the mesh coordinate vector. Symmetric constraints give constraint forces not only on the equations for $\mathbf{u}$ but also on the equations for $\mathbf{x}$, which typically are not wanted.
- Constraints on time derivatives, such as

$$\frac{\partial u}{\partial t} = 1$$

  on the boundary (typing `1-ut` using COMSOL syntax for R in the constraint $R = 0$). The default bidirectional symmetric constraint attempts to apply the test function on the time derivative of u, which is not supported. The solution is to apply the reaction terms on **Individual dependent variables**. Note that the constraint must also be a weak constraint because pointwise constraints for time derivatives are not supported.
- Wall boundary conditions for turbulent fluid flow. For the k-ε turbulence model this condition is of the type $k - r(\varepsilon), -\mathbf{n} \cdot \nabla\varepsilon$, where r is a given function. Bidirectional constraints for the first relation imply that the second relation cannot hold.

Unidirectional constraints can be enforced both in a pointwise sense and in a weak sense.

Turbulent fluid flow requires the CFD Module or Heat Transfer Module.

For descriptions about how to use even more general pointwise and weak nonsymmetric constraints, see Pointwise Constraint andWeak Constraint, respectively. Also see Boundary Condition Types.

The PDE Interfaces

COMSOL Multiphysics includes different PDE interfaces for equation-based modeling, distinguished by the equation formulation used for entering the equations—Coefficient Form, General Form, and Weak Form. The interfaces are identical except for the default node added to the top geometric entity level where the interface is active. You can still use a General Form to specify equations in a Coefficient Form PDE interface, or add Weak Form contributions to a General Form PDE interface.

The Wave Form PDE Interface is also available and described in another section.

- Modeling with PDEs
- Notational Conventions

Adding a PDE Interface to a Component

To add a new Component and use one of the equation interfaces, start with the instructions in Creating a New Model. Then, when you are adding the physics, expand the **Mathematics>PDE Interfaces** node in the list of physics interfaces and select one of the PDE interfaces in the list. For PDEs on other geometric entities than domains, expand the **Lower Dimensions** node.

SPECIFYING A SYSTEM OF EQUATIONS

COMSOL Multiphysics allows the creation of equations with more than one dependent variable. To do this, on the **Add Physics** page under **Dependent variables**, enter the **Number of dependent variables** in the field. COMSOL then automatically assigns variable names, typically `u1`, `u2`, `u3`, and so on. You can also edit the default variable name (as long as it is valid and unique) in the **Dependent variables** table. Several scalar PDEs can also be coupled using a multiphysics approach

For any form of PDE interface you add to a Component, additional equation nodes can be added in Coefficient Form, General Form, or Weak Form.

MODELING WITH PDES ON BOUNDARIES, EDGES, AND POINTS

The Coefficient Form PDE, General Form PDE, and Weak Form PDE are also available on boundaries, edges, and at points in the geometry.

Extra weak equations can be added by adding auxiliary dependent variables to a Weak Contribution (PDEs) node. Use such weak form equations as a way to handle thin layers; COMSOL then solves the problem by modeling rather than meshing. This approach reduces the solution time.

See Transport and Adsorption (model library path **COMSOL_Multiphysics/Chemical_Engineering/transport_and_adsorption**) to learn how to use the weak form boundary mode to model a thin adsorption layer with diffusion as a PDE on the boundary of a convection-diffusion problem.

See Shell Diffusion (model library path **COMSOL_Multiphysics/Equation-Based_Models/shell_diffusion**) for an example of tangential derivative variables.

- Modeling with PDEs
- Notational Conventions
- The PDE Interfaces

Discretization Section Shape Function Types and Element Orders

The PDE and weak form interfaces have different shape functions available with the associated element order (the order of the shape functions). Select the **Shape function type** and the **Element order** as, in most cases, **Constant**, **Linear**, **Quadratic**, **Cubic**, **Quartic**, or **Quintic** (order 0–5, respectively). Table 16-2 is an overview of the available shape function types and the element orders supported.

Not all shape functions are available for all space dimensions and types of equations, and not all shape functions support all orders.

TABLE 16-2: SHAPE FUNCTION TYPES

NAME	ORDER	COMMENTS
Lagrange	1–5 (1D and 2D); 1–4 (3D). Default: 2	The default type
Hermite	3–5 (1D and 2D); 3–4 (3D). Default: 3	
Argyris	Order 5 only.	2D only
Discontinuous Lagrange	0–5 (1D and 2D); 0–4 (3D). Default: 2	
Nodal discontinuous Lagrange	0–10	Special shape functions for wave equations
Discontinuous scalar density	0–5 (1D and 2D); 0–4 (3D). Default: 2	Not available on boundaries, edges, or points
Bubble	2 (1D); 3 (2D); 4 (3D)	Lower order on boundaries, edges, and points
Gauss point data	0, 2, 4, 8, or 10. Default: 4	Discrete values associated with the quadrature points in an integration rule of the given order

TABLE 16-2: SHAPE FUNCTION TYPES

NAME	ORDER	COMMENTS
Divergence	1–4 (1D and 2D); 1–3 (3D). Default: 2	For vector fields only (1, 2, and 3 dependent variables in 1D, 2D, and 3D, respectively
Curl	1–4 (2D); 1–3 (3D). Default: 2	2D and 3D only. For vector fields only (1, 2, and 3 dependent variables in 1D, 2D, and 3D, respectively)

The Coefficient Form PDE Interfaces

The **Coefficient Form PDE (c)** interface (Δu) found under the **Mathematics>PDE Interfaces** branch (Δu) when adding an interface, covers many well-known PDEs.

When this interface is added, these default nodes are also added to the **Model Builder—Coefficient Form PDE**, **Zero Flux**, and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **Coefficient Form PDE** to select physics from the context menu.

The Coefficient Form PDE interface is also available in other forms from the **PDE interfaces>Lower Dimensions** submenu—**Coefficient Form Boundary PDE**, **Coefficient Form Edge PDE**, and **Coefficient Form Point PDE**. Also see Modeling with PDEs on Boundaries, Edges, and Points.

The Coefficient Form PDE discusses the formulation and settings pertaining to the coefficient form, as well as the general PDE terminology used in COMSOL Multiphysics.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `c` (in domains), `cb` (on boundaries), `ce` (on edges), or `cp` (at points).

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

The default setting is to include **All domains**, **All boundaries**, **All edges**, or **All points** in the model. To choose specific geometric entities, select **Manual** from the **Selection** list.

UNITS

By default, the PDE interfaces are dimensionless, but units can be defined for the dependent variable and the source term (that is, the overall left and right side of the equation). The units for these quantities—in combination with the units for length and time—fully define the units for all other terms in the equations. Select the units from a list of physical quantities or enter the unit directly.

From the list, select the **Dependent variable quantity** that defines the unit for the dependent variable u. The default is **Dimensionless [1]**. Select **None** to enter a unit (for example, `K`, `m/s`, or `mol/m^3`) in the **Unit** field.

Select the **Source term quantity** that defines the unit for the source term f (the unit for the right—and left—side of the PDE). **None** is the default quantity, and `m^-2` is the default **Unit**, which is a consistent with a dimensionless dependent variable. Enter another unit (for example, `W/m^3` or `A/m^3`) in the **Unit** field as required.

For the **Classical PDE>Heat Equation** interface, the **Dependent variable quantity** defaults to **Temperature (K)** and the **Source term quantity** defaults to **Heat source (W/m^3)**.

DEPENDENT VARIABLES

Enter the **Number of dependent variables** (the default is 1) and set the field and dependent variable names. The default **Field name** and **Dependent variables** name for a single scalar PDE variable is u. If the **Field name** coincides with the name of another field of the same unit and number of components, the two fields (and the interfaces which define them) share degrees of freedom and dependent variable names.

A **Field name** must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**.

Select a **Shape function type** (finite element type)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (the order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

The **Compute boundary fluxes** check box is selected by default. Click to clear the check box as required. When you have selected to compute boundary fluxes, the **Apply smoothing to boundary fluxes** check box is available and selected by default. Click to clear the check box to turn off the smoothing.

The default **Value type when using splitting of complex variables** is **Complex**. Select **Real** if you want to use only real-valued data.

If you have added a Deformed Geometry or Moving Mesh interface, there is also a **Frame** list for specifying the frame for differentiation and quadrature. Choosing the frame can be needed if you want to write your own mesh smoothing or regularization equations when working with a deformed geometry. Select **Geometry**, **Mesh**, **Spatial** (the default), or **Material** from the **Frame** list.

- Modeling with PDEs
- Discretization Section Shape Function Types and Element Orders
- Computing Accurate Fluxes
- Compile Equations

For an example of the use of units in a PDE interface, see Shell Diffusion: model library path **COMSOL_Multiphysics/Equation-Based_Models/shell_diffusion**.

The General Form PDE Interfaces

The **General Form PDE (g)** interface (Δu), found under the **Mathematics>PDE Interfaces** branch (Δu) when adding an interface, is a flexible way to specify PDEs in a general form.

When this interface is added, these default nodes are also added to the **Model Builder**—**General Form PDE**, **Zero Flux**, and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **General Form PDE** to select physics from the context menu.

The General Form PDE interface is also available in other forms from the **PDE interfaces>Lower Dimensions** submenu—**General Form Boundary PDE**, **General Form Edge PDE**, and **General Form Point PDE**. Also see Modeling with PDEs on Boundaries, Edges, and Points.

The General Form PDE discusses the formulation and settings pertaining to the general form.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `g` (in domains), `gb` (on boundaries), `ge` (on edges), or `gp` (at points).

See The Coefficient Form PDE Interfaces remaining settings.

- General Form PDE
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs
- Modeling with PDEs

The Weak Form PDE Interfaces

The **Weak Form PDE (w)** interface, found under the **Mathematics>PDE Interfaces** branch (Δu) when adding an interface, are identical to The Coefficient Form PDE Interfaces and The General Form PDE Interfaces except for the default node on the top geometric entity level being a Weak Form PDE node.

The Weak Form PDE interface is also available in other forms from the **PDE interfaces>Lower Dimensions** submenu—**Weak Form Boundary PDE**, **Weak Form Edge PDE**, and **Weak Form Point PDE**. Also see Modeling with PDEs on Boundaries, Edges, and Points.

In all interfaces, weak expressions can be added, which COMSOL Multiphysics adds to the overall equation. Adding one of these interfaces creates a **PDE** node (∫du) for PDE modeling using a weak formulation. You can also add the same type of Weak Form PDE node on the domain level to any other PDE interface.

When this interface is added, these default nodes are also added to the **Model Builder**—**Weak Form PDE**, **Zero Flux** (for a Weak Form PDE on the domain level only), and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. On the domain level, edge level, and boundary levels the same boundary conditions can be used as for the Coefficient Form PDE and General Form PDE. You can also right-click **Weak Form PDE** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `w` (in domains), `wb` (on boundaries), `we` (on edges), or `wp` (at points).

See The Coefficient Form PDE Interfaces for the rest of the settings.

- Weak Form PDE
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs
- Modeling with PDEs

The Classical PDE Interfaces

Many classical PDEs are instances of Coefficient Form PDEs. The classical PDEs have their own interfaces found under the **Mathematics>Classical PDEs** branch (∇^2) when adding an interface. Classical PDEs can also be added to all of the forms of PDE interfaces as domain nodes.

The following Classical PDE interfaces and nodes are available. All of these have the same settings as The Coefficient Form PDE Interfaces:

- **Laplace Equation**
- **Poisson's Equation**
- **Wave Equation**
- **Helmholtz Equation**
- **Heat Equation**
- **Convection-Diffusion Equation**

- Compact and Standard Notations for Classical PDEs
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs

Domain, Boundary, Pair, Edge, and Point Conditions for PDEs

The PDE interfaces have the following domain, boundary, pair, edge, and point conditions described in this section and listed in alphabetical order. Some nodes are selected from the **Classical PDEs** submenu:

- Coefficient Form PDE
- Constraint
- Convection-Diffusion Equation
- Dirichlet Boundary Condition
- Flux/Source
- General Form PDE
- Heat Equation
- Helmholtz Equation
- Initial Values
- Laplace’s Equation
- Periodic Condition
- Poisson’s Equation
- Source, Edge Source, and Point Source
- Wave Equation
- Weak Form PDE
- Zero Flux

There are also auxiliary equation-based nodes found under the **More**, **Edges**, and **Points** submenus. To display these submenus in the context menu, click the **Show** button ( ) on the **Model Builder** toolbar and select **Advanced Physics Options**. Then choose from the following (listed in alphabetical order):

- Auxiliary Dependent Variable
- Discretization (Node)
- Pointwise Constraint
- Weak Constraint
- Weak Contribution (PDEs)
- Weak Contributions on Mesh Boundaries

There is generally a **More** submenu for the domain level as well as one for the boundary level on a physics context menu. See Physics Node Context Menu Layout for an example.

For some of the constraint nodes—Dirichlet Boundary Condition, Constraint, and Pointwise Constraint—you can add subnodes to exclude surrounding surfaces, edges, or points from the constraint. See Excluded Points, Excluded Edges, Excluded Surfaces.

- Modeling with PDEs
- The PDE Interfaces and Classical PDE Domain Nodes

Initial Values

The **Initial Values** node adds initial values for the dependent variables that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. If you need to specify more than one set of initial values, you can add additional **Initial Values** nodes.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the **Initial value for u**, u (dimensionless) and the **Initial time derivative of u** $\frac{\partial u}{\partial t}$ (SI unit: 1/s). The defaults are 0 for both dependent variables.

Coefficient Form PDE

The **Coefficient Form PDE** node is the default equation for The Coefficient Form PDE Interfaces, and is available for the other forms from the context menu. Specify the coefficients for a coefficient form PDE (see The Coefficient Form PDE and Equation 16-2)

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

DIFFUSION COEFFICIENT

Enter a value or expression for the diffusion coefficient c. Select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter a c coefficient on various forms in 2D and 3D. If there are multiple dependent variables, there is a matrix of c component inputs.

ABSORPTION COEFFICIENT

Enter a value or expression for the absorption coefficient a. If there are multiple dependent variables, there is a matrix of a component inputs.

SOURCE TERM

Enter a value or expression for the source term f. If there are multiple dependent variables, there is a vector of f component inputs.

MASS COEFFICIENT

Enter a value or expression for the mass coefficient e_a. If there are multiple dependent variables, there is a matrix of e_a component inputs.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping or mass coefficient d_a. If there are multiple dependent variables, there is a matrix of d_a component inputs.

CONSERVATIVE FLUX CONVECTION COEFFICIENT

Enter values or expressions for the conservative flux convection coefficient α vector's components. If there are multiple dependent variables, there is a matrix of α vector component inputs.

CONVECTION COEFFICIENT

Enter values or expressions for the convection coefficient β vector's components. If there are multiple dependent variables, there is a matrix of β vector component inputs.

CONSERVATIVE FLUX SOURCE

Enter values or expressions for the conservative flux source term γ vector's components. If there are multiple dependent variables, there is a vector of γ vector component inputs.

- Interpreting PDE Coefficients
- Specifying and Interpreting Boundary Conditions
- The PDE Interfaces
- Modeling with PDEs

General Form PDE

The **General Form PDE** node is the default equation for The General Form PDE Interfaces, and it is available for the other forms from the context menu. Specify the coefficients for a general form PDE (see The General Form PDE and Equation 16-1).

Except for Conservative Flux described in this section, see Coefficient Form PDE for the rest of the settings.

CONSERVATIVE FLUX

Enter values or expressions for the components of the conservative flux vector *Γ*. The default values `-ux`, `-uy`, and `-uz` (in 3D) represent the negative gradient of *u* and makes the left-hand side equal to the Laplace operator. If there are multiple dependent variables, there is one Γ vector for each variable.

- The PDE Interfaces
- Interpreting PDE Coefficients
- Modeling with PDEs

Weak Form PDE

The **Weak Form PDE** node is the default node on the top geometric entity level in The Weak Form PDE Interfaces and may also be added to The Coefficient Form PDE Interfaces and The General Form PDE Interfaces. It contains one weak form expression for each dependent variable in the interface (see Equation 16-9).

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

WEAK EXPRESSIONS

Enter the weak expressions that COMSOL Multiphysics (together with any other weak expressions on the same domain) sets equal to 0 in the **weak** field. For example, in a 2D Component model with one dependent variable, the default expression on the domain level is `-test(ux)*ux-test(uy)*uy+1[m^-2]*test(u)`. This is the weak formulation of Poisson’s equation with the right-hand side $f = 1$. On other geometric entity levels, the default weak expression is 0.

- The PDE Interfaces
- Modeling with PDEs

Source, Edge Source, and Point Source

You can add additional source term nodes on different geometry levels: **Source** on domains, **Edge Source** on edges (3D models), and **Point Source** at points.

DOMAIN, EDGE, OR POINT SELECTION

In the **Selection** list, add the geometric entities (domains, boundaries, edges, or points) to define as a source.

SOURCE TERM

Enter a value or expression for the source term f. The default is 0.

Classical PDE Domain Nodes

The nodes available from the **Classical PDEs** submenu can be added to any PDE interface at the domain level. The same node is also available as its own interface from the **Mathematics>Classical PDEs** branch (∇^2) when adding an interface.

See Coefficient Form PDE for all the settings and Compact and Standard Notations for Classical PDEs for the equations that the Classical PDE interface solve.

The available interfaces and domain nodes are:

LAPLACE’S EQUATION

The **Laplace Equation** is a classic PDE of elliptic type that can describe the behavior of some kind of potential or the steady-state heat equation.

POISSON’S EQUATION

The **Poisson’s Equation** is a classical PDE of elliptic type that can describe, for example, electrostatics with a space charge density.

HELMHOLTZ EQUATION

The **Helmholtz Equation** is a classical PDE of elliptic type that can represent, for example, a time-independent form of the wave equation.

WAVE EQUATION

The **Wave Equation** is a classic PDE of hyperbolic type. It is a second-order PDE that describes waves, such as sound waves, light waves, and water waves.

HEAT EQUATION

The **Heat Equation** is a classical PDE of parabolic type that describes time-dependent heat transfer by diffusion or other diffusion processes.

CONVECTION-DIFFUSION EQUATION

The **Convection-Diffusion Equation** is a classical PDE that describes time-dependent transport by convection and diffusion.

Dirichlet Boundary Condition

The **Dirichlet Boundary Condition** specifies a value of u on the boundary of the domain: $u = r$. By default, it is a unidirectional condition, applying reaction terms on u but not on any variables appearing in r.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose **Manual** (the default) to define the boundaries, edges, or points, or select **All boundaries**, **All edges**, or **All points** to select all applicable boundaries, edges, or points.

DIRICHLET BOUNDARY CONDITION

The Dirichlet boundary condition for each dependent variable (for example, u_2), has a corresponding check box (**Prescribed values for u2**), which is selected by default. Enter a value or expression for the prescribed value in the associated text field or click to clear the check box as required.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder**.

This section contains settings for specifying the type of constraint and whether to use a pointwise or weak constraint:

- From the **Apply reaction terms on** list, select **All physics (symmetric)** to apply reaction terms bidirectionally on u as well as on any other dependent variables appearing in r. Select instead **Current physics (internally symmetric)** to apply reaction terms on all components of u or **Individual dependent variables** to apply reaction terms from each active component in the Dirichlet condition only on the corresponding component of u. The latter is the default setting and effectively implies a unidirectional constraint.

Select the **Use weak constraints** check box to use weak constraints instead of the standard pointwise constraints.

You can add subnodes to exclude the constraint from any surrounding boundary, edge, or point. See Excluded Points, Excluded Edges, Excluded Surfaces.

See Coefficient Form PDE for all the settings and Compact and Standard Notations for Classical PDEs for the equations that the Classical PDE interface solve.

Constraint

The **Constraint** boundary condition specifies an expression R which is constrained to be equal to zero on the selection, $R = 0$. By default, this is a bidirectional constraint, meaning that all variables in R are affected by reaction terms.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose **Manual** (the default) to define the boundaries, edges, or points, or select **All boundaries**, **All edges**, or **All points** to select all applicable boundaries, edges, or points.

CONSTRAINT

Enter a value or expression for the value of R in the constraint $R = 0$. For example, to constrain u to 2, enter `2-u` in the field for R.

> The sign in front of u in the constraint controls the sign of the implicit Lagrange multiplier μ, as well as the sign of reaction forces computed using the `reacf()` operator. For consistency with the way predefined physics interfaces implement constraints, write `2-u` rather than `u-2` in R.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options** from the **Model Builder**. Normally these settings do not need to be changed.

This section contains settings for specifying the type of constraint and whether to use a pointwise or weak constraint:

- From the **Apply reaction terms on** list, select **All physics (symmetric)** to apply reaction terms bidirectionally on any dependent variables appearing in R. This is the default method for enforcing the constraint. Select instead **Current physics (internally symmetric)** to apply reaction terms only on components of u or **Individual dependent variables** to apply reaction terms from each constraint component only on the corresponding component of the dependent variable vector u.

Select the **Use weak constraints** check box to use weak constraints instead of the standard pointwise constraints.

> You can add subnodes to exclude the constraint from any surrounding boundary, edge, or point. See Excluded Points, Excluded Edges, Excluded Surfaces.

> See Coefficient Form PDE for all the settings and Compact and Standard Notations for Classical PDEs for the equations that the Classical PDE interface solve.

Excluded Points, Excluded Edges, Excluded Surfaces

Right-click a Constraint, Dirichlet Boundary Condition, or Pointwise Constraint node to add one or more **Excluded Points**, **Excluded Edges**, or **Excluded Surfaces** subnodes. Using those subnodes you can exclude all or part of the surrounding points, edges, or surfaces from a constraint that acts on the edge, boundary, or domain inside of the excluded geometric entities. For example, the **Excluded Edges** node specifies edges where the higher-dimensional (boundary) constraint for which it is a subnode is not enforced. To exclude a constraint on a surrounding edge can be useful to avoid that the constraint affects the physics on an adjacent boundary, for example.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose **Manual** to add the boundaries, edges, or points where you want to exclude the constraint or select **All boundaries** (for excluding surfaces in 3D and edges in 2D), **All edges** (for excluding edges in 3D), or **All points** (for excluding points) from the list to exclude the constraint from all applicable boundaries, edges, or points, respectively.

Flux/Source

The **Flux/Source** boundary condition adds a flux or source g on the boundary:

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)=g-qu \text{ or } -\mathbf{n}\cdot\Gamma = g-qu$$

The first equation describes this boundary condition for a Coefficient Form PDE, and the second term describes it for a General Form PDE. The g term may contain a general expression of the dependent variables. The q coefficient simplifies the implementation of a *Robin boundary condition* by including a term on the form qu, where u is the dependent variable.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose **Manual** (the default) to define the boundaries, edges, or points, or select **All boundaries**, **All edges**, or **All points** to select all applicable boundaries, edges, or points.

BOUNDARY FLUX/SOURCE

Enter a value or expression for the value of the boundary flux or source g in the corresponding field or fields. The default value is 0.

BOUNDARY ABSORPTION/IMPEDANCE TERM

Enter a value or expression for the value of the coefficient q in the corresponding field or fields. The default value is 0. It adds a term qu to the boundary flux or source, which can represent absorption or impedance at the boundary.

Zero Flux

The **Zero Flux** boundary condition is the default boundary condition and prescribes a zero flux (insulation) across the boundary:

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)=0 \text{ or } \mathbf{n}\cdot\Gamma=0$$

BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries, edges, or points to define or select **All boundaries**, **All edges**, or **All points** as required.

Periodic Condition

The **Periodic Condition** node adds a *periodic boundary condition*. This periodicity can be continuous (the default) so that $u(x_0)=u(x_1)$ or antiperiodic so that $u(x_0)=-u(x_1)$ and can control which dependent variables that the periodic condition applies to.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a periodic boundary condition. The software automatically identifies the boundaries as either source boundaries or destination boundaries. This works fine for cases like opposing parallel boundaries. In other cases, right-click **Periodic Condition** and add a Destination Selection subnode to control the destination. By default it contains the selection that COMSOL Multiphysics has identified.

💡 Also, in the **Model Builder** under **Component>Definitions**, a “read only” **Explicit** selection node is added and shows the selected destination boundaries.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** (the default) to make the dependent variables equal, or **Antiperiodicity** to make them antiperiodic: $u(x_0) = -u(x_1)$.

For each dependent variable in the PDE, choose to apply the periodic condition by selecting, for example, the **Apply condition on variable u1** check box. By default, the periodic condition applies to all dependent variables.

Periodic Boundary Conditions

Destination Selection

Right-click a Periodic Condition node to add the **Destination Selection** subnode and to change the selection for the destination. The selection that COMSOL Multiphysics makes appears as the default selection in the **Selection** list (as **Explicit 1**, for example).

BOUNDARY SELECTION

In the **Selection** list, add the boundaries to define as a destination. By default it contains the selection that COMSOL Multiphysics has identified.

The software usually automatically identifies the boundaries as either source boundaries or destination boundaries. By default it contains the selection that COMSOL Multiphysics identifies. A "read only" **Explicit** selection node ( ) (found in the **Model Builder** under **Component>Definitions**) shows the selected destination boundaries.

Periodic Boundary Conditions

Boundary Element PDE

The **Boundary Element PDE** node is available from the domain level **More** submenu in plane 2D and 3D PDE interfaces. It uses the boundary element method (BEM) to solve a source-free stationary PDEs with constant coefficients in selected domains. The solution is represented as an integral over all boundaries enclosing the domains, solving for the solution variable and numerical flux only on those boundaries.

To add this node, go to The Preferences Dialog Box. Click **Model Builder**, select the **Enable technology preview functionality** check box, and click **OK**.

The node can then be added by right-clicking the mathematics PDE interface node (for example, **Coefficient form PDE**) and selecting the option from the domain level **More** submenu.

About Boundary Element Methods

DOMAIN SELECTION

Add the domain where you want to solve the PDE to the **Selection** list, select **All domains**, or choose a named domain selection, when available.

DOMAIN SELECTION

From the **Selection** list, choose the domains or select **All domains** from the list. The node represents PDE defined by its constant coefficients in the selected domains.

DIFFUSION COEFFICIENT

Enter a numerical value or a constant expression for the isotropic diffusion coefficient c. If there are multiple dependent variables, there is a list of component inputs for diagonal elements of the general c component matrix. All nondiagonal elements are implicitly set to zero.

ABSORPTION COEFFICIENT

Enter a value or a constant expression for the absorption coefficient a. If there are multiple dependent variables, there is a list of component inputs for diagonal elements of a. All nondiagonal elements are implicitly set to zero.

CONSERVATIVE FLUX CONVECTION COEFFICIENT

Enter values or constant expression for the conservative flux convection coefficient α vectors components. If there are multiple dependent variables, there is a list of α vectors components inputs for diagonal elements. All nondiagonal elements are implicitly set to zero

CONVECTION COEFFICIENT

Enter values or a constant expressions for the convection coefficient β vectors components. If there are multiple dependent variables, there is a list of β vectors component inputs for diagonal elements of β. All nondiagonal elements are implicitly set to zero.

OPERATOR NAME

The **Boundary Element PDE** feature solves the PDE in the selected domains using dependent variables and degrees of freedom which are defined only on the exterior boundaries. Inside the domains the solution is evaluated using an operator accepting coordinates as arguments. The default **Operator name** for a dependent variable u is `bemop_u.` The full operator name, which must be used in postprocessing, includes the feature scope, for example, `c.bempde1.bemop_u(x,y,z)`. In case of multiple dependent variables there is a list of operator name inputs for all the dependent variables.

> Evaluation of the operator can be expensive because for each evaluation point it involves integration over all the boundaries of the node selection. The evaluation of the operator is not accurate in the vicinity of the boundaries of the node selection.

QUADRATURE SETTINGS

Enter values or expressions for each of the following dimensionless numbers specifying integration order for double integral contributions from different types pairs of element or for single integral contribution (default values shown here assume second-order shape functions used for discretization of dependent variables):

- **Integration order, distant elements**. Integration order for contributions to double integrals from pairs of distant elements. The default is 4 in 2D geometry and 2 for 3D.
- **Integration order, close elements**. Integration order for contributions to double integrals from pairs of close elements. The default is 8 in 2D geometry and 4 in 3D.

- **Integration order, elements with common vertex**. Integration order for contributions to double integrals from pairs of elements sharing common vertex. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, elements with common edge**. Integration order for contributions to double integrals from pairs of elements sharing common edge. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, same element**. Integration order for contributions to double integrals from pairs of same elements. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, weak contribution**. Integration order for contributions to single integrals. The default is 4.

All the integration orders for double integrals should be at least equal to the discretization order of the dependent variable, but for pairs of close or overlapping elements, higher orders are recommended. The integration order for single integrals (**Integration order, weak contribution**) should be at least twice the discretization order of the dependent variable.

BOUNDARY FLUX DISCRETIZATION

The only supported **Shape function type** for boundary element flux is **Lagrange** shape functions. Select an associated **Element order** (the order of the shape function for the element). The default is to use order one less than the discretization order for the dependent variable or 1, if the discretization order for dependent variable is 1.

Exterior Boundary Element PDE

The **Exterior Boundary Element PDE** node is available from the domain level **More** submenu in plane 2D and 3D PDE interfaces. It uses the boundary element method (BEM) to solve a source-free stationary PDEs with constant coefficients in a region exterior to the PDE interface domain selection. This region can include closed inner cavities as well as the outer infinite region. The solution is represented as an integral over a selected set of closed boundaries, solving for the solution variable and numerical flux only on those boundaries..

To add this node, go to The Preferences Dialog Box. Click **Model Builder**, select the **Enable technology preview functionality** check box, and then click **OK**.

The node can then be added by right-clicking the mathematics PDE interface node (for example, **Coefficient form PDE**) and selecting the option from the boundary level **More** submenu.

About Boundary Element Methods

BOUNDARY SELECTION

Add the boundaries where you want to solve the PDE to the **Selection** list, select **All boundaries**, or choose a named boundary selection, when available. The node represents a PDE defined by its constant coefficients in the region (possibly infinite) or regions adjacent to the selected boundaries that are exterior to the parent physics interface. Only exterior boundaries can be selected and the selection has to form a closed boundary or a set of closed boundaries.

DIFFUSION COEFFICIENT

Enter a value or a constant expression for the isotropic diffusion coefficient c. If there are multiple dependent variables, there is a list of component inputs for diagonal elements of the general c component matrix. All nondiagonal elements are implicitly set to zero.

ABSORPTION COEFFICIENT

Enter a value or a constant expression for the absorption coefficient a. If there are multiple dependent variables, there is a list of component inputs for diagonal elements of a. All nondiagonal elements are implicitly set to zero.

CONSERVATIVE FLUX CONVECTION COEFFICIENT

Enter values or constant expression for the conservative flux convection coefficient α vectors components. If there are multiple dependent variables, there is a list of α vectors components inputs for diagonal elements. All nondiagonal elements are implicitly set to zero.

CONVECTION COEFFICIENT

Enter values or a constant expressions for the convection coefficient β vectors components. If there are multiple dependent variables, there is a list of β vectors component inputs for diagonal elements of β. All nondiagonal elements are implicitly set to zero.

CONDITION AT INFINITY

If the exterior region defined by the boundary selection of the **Exterior Boundary Element PDE** node is an infinite domain, the solutions of the Helmholtz and Laplace equations are not uniquely defined without an additional condition.

Select the **Condition at infinity** — **None** (the default, to be used for bounded domains), **Helmholtz equation**, or **Laplace equation** depending on the character of the exterior region defined by the boundary selection and the constant coefficients used for the PDE:

- Choose **None** (default) as condition at infinity if the exterior region adjacent to the selected boundaries is finite, or if the constant coefficients used in the node specify neither a Helmholtz equation nor a Laplace equation.
- Choose **Helmholtz equation** if the constant coefficients used in the node specify a Helmholtz equation and the exterior region adjacent to the selected boundaries is infinite. In this case select the wave type — **Outgoing wave** (default), **Incoming wave**, or **General**. If **General** is chosen, enter a value for m (dimensionless) to specify a general combination of outgoing and incoming waves. The default is 0 (resulting in real-valued solutions).
- Choose **Laplace equation** if the constant coefficients used in the node specify the Laplace equation and the exterior region adjacent to the selected boundaries is infinite. In this case choose whether to supply an **Asymptotic value at infinity** (default) or the **Total flux through boundary**. If **Asymptotic value at infinity** is chosen, enter an asymptotic value for each of the dependent variables in the interface. The default value is 0. If **Total flux through boundary** is chosen, enter the desired total flux of each of the dependent variables through the selected boundaries. In this case additional degrees of freedom for the value at infinity are added internally, one for each dependent variable in the interface.

Because the solution of Laplace equation in 2D has asymptotic behavior in the form `a.log(r)+b,` the asymptotic value at infinity refers in this case to the constant additive term `b`.

OPERATOR NAME

The **Exterior Boundary Element PDE** feature solves the PDE in an exterior virtual domain using dependent variables and degrees of freedom which are defined only on boundaries. In order to evaluate the solution in the exterior region, which may not be meshed at all, you must use an operator accepting coordinates as arguments. The default **Operator name** for a dependent variable u is `bemop_u.` The full operator name, which must be used in postprocessing, includes the feature scope, for example, `c.bempde1.bemop_u(x,y,z)`. In case of multiple dependent variables there is a list of operator name inputs for all the dependent variables.

Evaluation of the operator can be expensive because for each evaluation point it involves integration over all the boundary selection. The evaluation of the operator is not accurate in the vicinity of the boundary selection.

QUADRATURE SETTINGS

Enter values or expressions for each of the following dimensionless numbers specifying integration order for double integral contributions from different types pairs of element or for single integral contribution (default values shown here assume second-order shape functions used for discretization of dependent variables):

- **Integration order, distant elements**. Integration order for contributions to double integrals from pairs of distant elements. The default is 4 in 2D geometry and 2 for 3D.
- **Integration order, close elements**. Integration order for contributions to double integrals from pairs of close elements. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, elements with common vertex**. Integration order for contributions to double integrals from pairs of elements sharing common vertex. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, elements with common edge**. Integration order for contributions to double integrals from pairs of elements sharing common edge. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, same element**. Integration order for contributions to double integrals from pairs of same elements. The default is 8 in 2D geometry and 4 in 3D.
- **Integration order, weak contribution**. Integration order for contributions to single integrals. The default is 4.

All the integration orders for double integrals should be at least equal to the discretization order of the dependent variable, but for pairs of close or overlapping elements, higher orders are recommended. The integration order for single integrals (**Integration order, weak contribution**) should be at least twice the discretization order of the dependent variable.

BOUNDARY FLUX DISCRETIZATION

The only supported **Shape function type** for boundary element flux is **Lagrange** shape functions. Select an associated **Element order** (the order of the shape function for the element). The default is to use order one less than the discretization order for the dependent variable or 1, if the discretization order for dependent variable is 1.

About Boundary Element Methods

BACKGROUND

Boundary element methods provide an alternative to the finite element methods in case of a source free stationary PDE with constant coefficients:

$$\nabla \cdot (-c\nabla u - \alpha u) + \beta \cdot \nabla u + a = 0$$

The solution of such PDE can be expressed in terms of integrals of the fundamental solutions over the boundary of the domain. There are various possible formulations of the boundary element method. The boundary element methods in COMSOL use a direct method, where the solution of the PDE inside the domain is expressed directly in terms of the field and field flux on the domain boundary. The field flux on the boundary is defined as $\mathbf{n}(c\nabla u + \alpha u)$, where $\mathbf{n}$ is the outward normal vector. This formulation allows easy handling of the boundary conditions, which can be specified in the same way as boundary conditions for the finite element methods, and it also make it straightforward to connect domains represented with the boundary element methods to the domains represented by the finite element methods.

In boundary element methods there is no need to mesh and discretize the entire domain where you want to solve the PDE (which could even be an infinite exterior space). This typically reduces the number of degrees of freedom because it requires discretization of the field and field flux only at the boundaries. However, unlike the matrices from finite element methods, which are sparse, the resulting matrices from boundary element methods are dense, which can result in higher memory consumption.

Assembling of the matrices for the boundary element method involves double integrals of singular functions (fundamental solutions) over the domain boundary. Therefore, special integration formulas have to be used during the assembling in order to handle the singularities. Even with the special integration formulas used, higher integration orders are needed for contributions from pairs of overlapping elements where singularities play a bigger role than from pairs of distant elements.

Reference

For more details about the boundary element methods, see S.A. Sauter and C. Schwab, *Boundary Element Methods*, Springer-Verlag, 2011.

USING BOUNDARY ELEMENT METHODS

Available Boundary Element PDE Nodes

Boundary element methods are provided by two equation-based modeling features in COMSOL: The **Boundary Element PDE** and **Exterior Boundary Element PDE** nodes. These nodes are available s subnodes of the 2D or 3D Mathematics PDE interfaces (for example, a Coefficient Form PDE interface) when you have enabled the **Enable technology preview functionality** preference.

- The **Boundary Element PDE** is defined on a domain selection and can be used to represent stationary source-free PDEs with constant coefficients in the domains or the whole selection of the parent domain's physics selection. This node overrides any other PDE-defining node that is defined on any of the selected domains and disables the discretization of the dependent variables inside the selected domains. No nodes that contribute to the PDE inside the domains (like, for example, sources) are allowed.
- The **Exterior Boundary Element PDE** is defined through boundary selection on the exterior boundaries of the parent physics interface. The selected boundaries have to form a closed boundary or a set of closed boundaries in order to be a valid selection for this feature. Using this feature can be useful if there is a need to include a PDE in an infinite domain into the model.
- For both the **Boundary Element PDE** and the **Exterior Boundary Element PDE** it is possible to combine them with adjacent finite element PDE features or to specify Dirichlet or flux constraints on the boundaries in exactly the same way as they are used for finite element PDEs. For the **Exterior Boundary Element PDE**, if it represents a Helmholtz or Laplace equation in an infinite domain, there is an additional constraint needed in order to uniquely define the solution. These conditions are specified in the **Condition at Infinity** section. For the Helmholtz equation this means specifying whether the solution is an outgoing or incoming wave or a general combination of the two. For the Laplace equation this condition can be specified either by specifying asymptotic value at infinity (value of the additive constant term in 2D) or total flux through the boundary.

If the physics interface for which the boundary element features are used has multiple field components (dependent variables), the boundary element features represent uncoupled PDE equations, which means that there is one separate PDE equation for each dependent variable.

Integration Order

Because the integrals evaluated by the boundary element method involve double integrals over singular functions, it is desirable to use a higher integration order for contributions from pairs of elements that are close or even overlap (identical elements with a common edge or vertex). Therefore the features allow setting the integration order for these categories separately under the **Quadrature Settings** section. The integration order for the all contributions from pairs of elements should be at least equal to the discretization order used for the dependent variable. For contributions from the pairs of distant elements, the integration order that is the same as the discretization order used for the dependent variables is typically sufficient. For contributions from pairs of close elements and overlapping elements, a higher order is usually needed. For terms at the boundary that are assembled in the form of single integral over boundaries in a similar way as terms appearing in the finite element formulation, you specify the integration order using the **Integration order, weak contribution** setting. This integration order should be set to the integration order that would be used if finite element features were used for the problem (typically twice the order of the dependent variables).

The boundary element features create an auxiliary dependent variable at the boundary to represent the flux. The order of discretization for this variable can be set in the **Boundary flux discretization** section. Typically the order is set to one less than the order of the dependent variable discretization or to 1 if the dependent variable discretization order is 1.

Evaluation of Field Variables in Domains

Because boundary element nodes do not provide shape functions for the field variables in the domains where they represent the PDE, they provide an alternative way of evaluation of the field variables in these domains. For each dependent variable it provides an operator that can be evaluated at any point in space (although the only relevant use of these operators is inside the domains where the boundary element nodes represent their PDEs). As there can exist multiple boundary element nodes for a single physics interface, these operators have a scope specific for the node that defined them, for example, `comp1.c.bempde1.bemop_u(x,y,z)`. You find the operator name and scope in the **Operator Name** section of the boundary element node's Settings window. Each boundary element node also defines variables in terms of these operators, which can be used during postprocessing, either on domains where the boundary element PDE is defined, or in case of an **Exterior Boundary Element PDE** as global variables. These variables also have feature scope (for example, `comp1.c.bempde1.bem_u`), and you can choose them in the list of predefined quantities for plotting and evaluation. For all operators and variables, their spatial derivatives are also available..

Because evaluation of these operators and variables involves, for each evaluation point, an integration over boundaries, the evaluation of these operators and variables is much more time consuming than the evaluation of variables discretized with finite element methods.

Due to the presence of singular integrands, the fields and field gradients evaluated in this way are not accurate in the vicinity of or at the boundaries.

Theory for the Wave Form PDE

The Wave Form PDE Interface theory is described in this section.

Derivation of the Weak Form of the Wave Form PDE

Using The Wave Form PDE Interface it is possible to solve one or several first-order wave equations; that is, PDEs of the form

$$d_a\frac{\partial u}{\partial t} + \nabla \cdot \Gamma(u) = f$$

(16-10)

where u is the unknown, d_a the mass coefficient, f the source, and Γ the flux vector, which generally depends on u.

The numerical method consists of a discontinuous Galerkin (DG) method in space in combination with explicit Runge-Kutta time stepping. This combination of space and time discretization is particularly well-suited for wave problems due to a favorable CFL condition. This is true even when using a high-order polynomial ansatz for u.

In order to derive the weak form underlying the DG method, let $\{\Omega^e\}$ denote a mesh of the domain $\{\Omega\}$, with Ω^e denoting a single element. On this mesh, let V be the broken space

$$V = \oplus P^s(\Omega^e)$$

with P^s denoting the space of all polynomials of degree at most s on Ω^e.

These functions are continuous in the interior of each mesh element, but generally discontinuous across the element boundaries.

A basis for V is given by the nodal discontinuous Lagrange shape functions `shhwdisc.` This basis is tailor made for this type of DG method, and it has nearly optimal interpolation properties.

The starting point for deriving a weak form is to multiply the PDE with a test function $v \subset V$ and integrate over the domain, to yield

$$\int_{\Omega} d_a\frac{\partial u}{\partial t} v dA + \int_{\Omega} \nabla \cdot \Gamma v dA = \int_{\Omega} f v dA$$

The next step is to integrate by parts. Some care must be taken, since the integrands v and u are discontinuous functions across element boundaries and only continuous in the interior of each element. Therefore, the integrals are first written as a sum over the elements and then integration by parts is done on each element, which gives

$$\sum_e \int_{\Omega^e} d_a\frac{\partial u}{\partial t} v dA - \sum_e \int_{\Omega^e} (\Gamma \cdot \nabla v) dA + \sum_e \int_{\partial\Omega^e} (\mathbf{n} \cdot \Gamma^*) v ds = \sum_e \int_{\Omega^e} f v dA$$

where $\mathbf{n}$ is the outward unit normal on the element. Further, Γ^* is the so-called numerical flux, which defines the flux vector on each element boundary. The flux vector is usually discontinuous because it depends on u.

The numerical flux defines how adjacent elements are connected and how continuous u is. Different definitions of the numerical flux lead to different variants of DG methods.

The numerical flux implemented in COMSOL Multiphysics is the global Lax-Friedrichs flux:

$$\mathbf{n}\cdot\Gamma^* = \mathbf{n}\cdot\langle\Gamma\rangle + \tau[u]$$

where the angles $\langle\ \rangle$ and brackets $[\ ]$ are the average and jump operators, respectively. Thus, on each element boundary this flux is simply the average of the flux on the two adjacent elements sharing the face, plus a penalty on any jumps of the solution. The penalty is needed for stability and is proportional to the parameter τ, which is assumed to be constant over the whole domain Ω. Using the definition of the Lax-Friedrichs flux, the weak form is obtained

$$\sum_{e}\int_{\Omega^e} d_a\frac{\partial u}{\partial t}v\,dA - \sum_{e}\int_{\Omega^e}(\Gamma\cdot\nabla v)dA + \sum_{e}\int_{\partial\Omega}(\mathbf{n}\cdot\langle\Gamma\rangle + \tau[u])v\,ds = \sum_{e}\int_{\Omega} fv\,dA$$

What makes this DG method particularly attractive for explicit time stepping is the fact that the term

$$d_a\frac{\partial u}{\partial t}$$

yields a block diagonal mass matrix, where each block only involves the degrees of freedom on each element. As a consequence, there is no need to invert, or solve any linear system involving the global mass matrix. The inverse of the global mass matrix simply amounts to inverting the local mass matrix on each element. This is efficiently done using high performance routines such as BLAS or LAPACK.

A known drawback with explicit time stepping is the requirement on the time step, which has to be very small in order to obtain a stable numerical method. This is referred to as the CFL condition, which relates the largest possible time step k to the smallest mesh size h. For wave equations with unit wave speed, the CFL condition takes the form

$$k \le C\frac{h}{\mathrm{p}^2}$$

where p is the order of the shape functions and C a generic constant, typically 0.25.

As implemented in COMSOL, the *nodal discontinuous Lagrange shape functions* are the only set of shape functions defined for this interface. The associated element order can be chosen from the **Element order** list. The highest available order is four, and the default order is two.

The nodal discontinuous Lagrange shape functions can only be defined on triangular meshes in 2D or tetrahedral meshes in 3D.

See Discretization as defined for The Wave Form PDE Interface.

Time Explicit Integrator

After discretization in space, an explicit ordinary system of differential equations is obtained. The standard procedure is to integrate the DG system of equations with the explicit Runge-Kutta family of methods.

The combination is in the literature denoted RK-DG. Often, a p+1 order RK method is combined with p:th order shape functions. See Ref. 1 for details. The new integrator supports a CFL-based time step regulator. The stability limitation for the time step k is of the sort

$$k \leq \frac{C}{(1+2\mathrm{p})^2}\min\frac{h}{\left|\lambda\left(\frac{\partial\Gamma}{\partial u}\right)\right|}$$

where C is a moderate constant.

COMSOL Multiphysics can compute the largest time step k, based on the smallest mesh size $h,$ the order of the shape functions, and the maximum wave speed in the domain. Under **Time stepping**, select **From expressions** to automatically compute the largest stable time step k. Under **Cell time scale expressions** add `wahw.wtc` as the variable for the **Estimate of Maximum Wave Speed** when using The Wave Form PDE Interface.

- Time Explicit Solver
- Introduction to Solvers and Studies

Local Time Stepping

The time step restriction is directly proportional to the smallest mesh element size. The CFL condition is normally too restrictive on highly graded meshes. On such a mesh it is common that only few elements are small, yet these dictate the overall time step for the whole problem. As an example, think of a mesh with a small geometric feature somewhere or a mesh stemming from an adaptive computation.

In these cases one option is to use local time stepping, which allows the use of a larger time step based on the size of the majority of the elements. This is possible due to the element-wise nature of the DG scheme. The larger the spread is between the smallest mesh-element size and the ideal mesh-element size dictated by a points per wavelength argument, the more beneficial this technique is.

The basic problem with local time stepping, aside from stability issues, is to obtain high accuracy. Classical results only involve second order accuracy. This is not good enough for DG methods.

In order to perform local time stepping, the third order classical Adams-Bashforth (AB) method is implemented

$$u_{n+1} = u_n + \frac{k}{12}(23R(u_n) - 16R(u_{n-1}) + 5R(u_{n-2})) \qquad (16\text{-}11)$$

here, u_n is the solution at time t_n, k is the time step, and $R(u_n)$ the weak form $f - \nabla \cdot \Gamma$ of the DG scheme (compare to the weak formulation of Equation 16-10), which includes the numerical flux.

The basic idea with local time stepping is as follows. First, the elements are divided into groups, based on their size. Typically, the groups are constructed such that the time steps k, $k/2$, $k/4$, and so forth, are stable time steps for each group of elements and where any coupling between groups is disregarded. This allows for easy synchronization of the solution at every full step. The right-hand-side vectors

$$R(u_n)\text{, } R(u_{n-1})\text{, and } R(u_{n-2})$$

are expensive to compute, thus these are naturally stored from previous time steps, so for each group of DOFs, their own history of right-hand sides are stored.

The main idea with local time stepping is to match the different groups with their own time step and thus save computational resources.

- Time Explicit Solver
- Introduction to Solvers and Studies

Reference for the Wave Form PDE Interface

1. Jan S. Hesthaven and Tim Warburton, *Nodal Discontinuous Galerkin Methods— Algorithms, Analysis, and Applications*, Springer, 2008.

The Wave Form PDE Interface

You enable this interface in The Preferences Dialog Box. Click **Model Builder**, and select the **Enable technology preview functionality** check box; then click **OK**.

The **Wave Form PDE** interface (Δu), found under the **Mathematics>PDE Interfaces** branch (Δu) when adding an interface, solves wave equations formulated with first order derivatives in time and space using the discontinuous Galerkin method and is highly optimized with respect to speed and memory consumption.

This section covers the formulation and settings pertaining to those equations.

When you add this interface, these default nodes are also added to the **Model Builder**—**Wave Form PDE**, **Zero Flux**, and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **Wave Form PDE** to select physics from the context menu.

Use the *wave form* for first-order hyperbolic PDEs. Assuming a scalar equation for the dependent variable u, these problems take the form

$$\begin{aligned} d_a\frac{\partial u}{\partial t} + \nabla\cdot\Gamma = f \qquad & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = g \qquad & \text{on}\,\partial\Omega \end{aligned}$$

together with suitable initial data. The first equation is the PDE, the second the boundary conditions.

The terms d_a, Γ, *f*, and *g* are coefficients. They can be functions of both the spatial coordinates or time, and the solution u, but not the derivatives of u. The coefficients *f* and *g* are scalar, whereas Γ is the *flux vector*. The coefficient d_a is assumed to be nonzero throughout the domain Ω, and for all times.

The **Wave Form PDE** interface also supports systems of equations. The interpretation of the coefficients are the same as for the scalar case; *f* and *g* are vectors with one component for each equation. Γ contains one flux vector for each equation, and d_a is a square (invertible) matrix.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `wahw`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

UNITS

Select a **Dependent variable quantity**. **Dimensionless (1)** is the default and there are many other options to choose from. Select a **Source term quantity**. **None** is the default and there are many other options to choose from. Enter a **Unit**.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**.

Due to efficiency reasons, there is only one **Shape function type** (finite element type) defined for this interface—the **Nodal discontinuous Lagrange** functions. The associated element order (the order of the shape function for the element) can be chosen from the **Element order** list. The highest available order is 10 in 1D and 2D models and 4 in 3D models, and the default order is two.

> The Nodal Discontinuous Lagrange shape functions can only be defined on triangular meshes in 2D, or tetrahedral meshes in 3D.

DEPENDENT VARIABLES

Define the number of the dependent variables and the variable names. The default name for a scalar PDE variable is u.

> ! The dependent variables and their names must be unique with respect to all other dependent variables in the model.

Add or remove dependent variables in the model and also change their names.

Enter the **Number of dependent variables**. Use the **Add dependent variable** (+) and **Remove dependent variable** ( ) buttons as required.

- Domain and Boundary Physics for the Wave Form PDE Interface
- Show More Physics Options
- Theory for the Wave Form PDE
- The Time Explicit Solver Algorithms
- Using Units

Domain and Boundary Physics for the Wave Form PDE Interface

The Wave Form PDE Interface includes the following domain and boundary physics nodes:

- Flux/Source
- Initial Values
- Interior Flux
- Interior Source
- Wave Form PDE
- Zero Flux

Wave Form PDE

This is the default equation for a **Wave Form PDE** interface. Here the coefficients for a wave form PDE are specified with the following equation coefficients:

$$d_{\mathrm{a}}\frac{\partial u}{\partial t} + \nabla \cdot \Gamma(u) = f$$

- d_{a} is the mass coefficient
- $\Gamma(u)$ is the conservative flux vector
- f is the source term

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping (or mass) coefficient d_{a} (SI unit: s/m^2). The default is 1 s/m^2. If there are multiple dependent variables, there is a matrix of d_{a} component inputs.

CONSERVATIVE FLUX

Enter values or expressions for the components of the conservative flux vector $\Gamma(u)$ (SI unit: 1/m). These components may depend on both the spatial and temporal coordinates, and the solution u, but not any derivatives of u. If there are multiple dependent variables, there is one $\Gamma(u)$ vector for each variable.

SOURCE TERM

Enter a value or expression for the source term f (SI unit: 1/m^2). If there are multiple dependent variables, there is a vector of f component inputs. The default is 0 (1/m^2).

LAX-FRIEDRICHS FLUX PARAMETER

The shape functions used are discontinuous and therefore require auxiliary constraints on faces between adjacent mesh elements to yield a meaningful (that is, continuous) solution approximation. This is accomplished by specifying a so-called numerical flux on each face. The numerical flux implemented is the (global) Lax-Friedrichs flux, which is defined as the average of the fluxes on neighboring elements plus the jump of the solution times a parameter τ, which is necessary for stability.

Enter a value or global expression for the parameter τ. Only one expression can be entered for each equation and each domain. The parameter is per default one, but should be set accordingly to the dominant eigenvalue of the flux Jacobian matrix

$$\bar{\lambda} = \max\left|\lambda\left(d_a^{-1}\left(\mathbf{n}\cdot\frac{\partial\Gamma}{\partial u}\right)\right)\right| \tag{16-12}$$

given the bound

$$0 \le \tau \le \frac{\bar{\lambda}}{2} \tag{16-13}$$

A so-called *central flux* is obtained for $\tau = 0$. This gives a non-dissipative method and gives an approximation that does not penalize jumps in between elements.

Selecting $\tau = \frac{\bar{\lambda}}{2}$ sets a maximally dissipative flux, which corresponds to a so-called *upwind flux* for smoothly varying parameters $d_a, \frac{\partial\Gamma}{\partial u}$.

Lax-Friedrichs Flux Parameter

ESTIMATE OF MAXIMUM WAVE SPEED

Enter a value or expression for the estimate of maximum wave speed W_s. The default is 0.

FILTER PARAMETERS

The filter provides higher-order smoothing of nodal discontinuous Galerkin formulations and is intended to be used for absorbing layers, but you can also use it to stabilize linear wave problems with highly varying coefficients. The filter is constructed by transforming the solution (in each global time step) to an orthogonal polynomial representation, multiplying with a damping factor and then transforming back to the (Lagrange) nodal basis. Select the **Activate** check box to use this filter.

The exponential filter can be described by the matrix formula

$$V\Lambda V^{-1}$$

where V is a Vandermonde matrix induced by the node points, and Λ is a diagonal matrix with the exponential damping factors on the diagonal:

$$\Lambda_{mm} = \sigma(\eta) = \begin{cases} 1, 0 \le \eta \le \eta_c \\ e^{-\alpha\left(\frac{\eta - \eta_c}{1 - \eta_c}\right)^{2s}}, \eta_c \le \eta \le 1 \end{cases}$$

where

$$\eta = \eta(m) = \frac{i_m}{N_p}$$

and N_p is the basis function and i_m the polynomial order for coefficient m. α (default value: 36), η_c (default value: 0.6), and s (default value: 3) are the filter parameters that you specify in the corresponding text fields. The damping is derived from an a spatial dissipation operator of order $2s$. For $s = 1$, you obtain a damping that is related to the classical 2nd-order Laplacian. Higher order (larger s) gives less damping for the lower-order polynomial coefficients (a more pronounced low-pass filter), while keeping the damping property for the highest values of η, which is controlled by α. The default values 36 for a correspond to maximal damping for $\eta = 1$. It is important to realize that the effect of the filter is influenced by how much of the solution (energy) is represented by the higher-order polynomial coefficients. For a well resolved solution this is a smaller part than for a poorly resolved solution. The effect is stronger for poorly resolved solutions than for well resolved ones. This is one of the reasons why this filter is useful in an absorbing layer where the energy is transferred to the higher-order coefficients through a coordinate transformation. See Ref. 1 (Chapter 5) for more information.

α must be positive; $\alpha = 0$ means no dissipation, and the maximum value is related to the machine precision, $-\log(\varepsilon)$, which is approximately 36. η_c should be between 0 and 1, where $\eta_c = 0$ means maximum filtering, and $\eta_c = 1$ means no filtering, even if filtering is active.

Initial Values

The **Initial Values** node adds the initial values for the dependent variables to be specified. These serve as an initial condition for the transient simulation.

DOMAIN SELECTION

In the **Selection** list, add the domains where the initial value is active.

> If there is more than one type of domain, each with different initial values defined, it may be necessary to remove these domains from the selection. These are then defined in an additional **Initial Values** node.

INITIAL VALUES

Enter a value or expression for the **Initial value for u** u (dimensionless). The default value is 0.

Zero Flux

The **Zero Flux** boundary condition is the default boundary condition and prescribes a zero flux across the boundary:

$$\mathbf{n} \cdot \Gamma = 0 \, .$$

It can be used on exterior boundaries only.

Flux/Source

The **Flux/Source** boundary condition can be used on exterior boundaries only. With this node the boundary condition is enforced according to:

$$-\mathbf{n} \cdot \Gamma = g + qu$$

where g and q can be specified. Here the dependent variable u in the right-hand side is evaluated on the inside (as seen from the domain where the PDE interface is defined). When the normal vector is used in expressions for curved boundaries, it is important that the mesh version of these vectors are used. The components of this vector are denoted `nxmesh`, `nymesh`, and `nzmesh`, respectively.

> In this context, exterior means exterior to the PDE interface, which does not need to be an exterior boundary to the geometry. On such boundaries, the normal direction as defined by the normal vector variables does not necessarily has to be outward pointing.

BOUNDARY SELECTION

In the **Selection** list, add the boundaries to define as a boundary flux or source.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

BOUNDARY ABSORPTION/IMPEDANCE TERM

Enter the flux term q (SI unit: 1/m). The default is 0 (1/m).

Interior Source

The **Interior Source** boundary condition can be used on interior boundaries only. With this node the boundary condition is enforced according to:

$$-\mathbf{n} \cdot \Gamma = -\mathbf{n} \cdot \Gamma_{\mathrm{LF}} + g$$

where Γ_{LF} is the Lax-Friedrichs flux computed internally. Here the source g can be specified.

BOUNDARY SELECTION

In the **Selection** list, add the interior boundaries to define as a source.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

Interior Flux

The **Interior Flux** boundary condition can be used on interior boundaries only. With this node the boundary condition is enforced according to:

$$-\mathrm{down}(\mathbf{n})\cdot\Gamma = g \text{ and } -\mathrm{up}(\mathbf{n})\cdot\Gamma = -g$$

where g can be specified. It is evaluated on the interior boundary, where `up` and `down` operators are supported. If a dependent variable is used in this expression (without `up` or `down` operators, an implicit mean operation is invoked taking the average of the up and down values).

To appreciate how this boundary condition works, consider a simple example of a one-way transport equation in 1D

$$u_t + au_x = 0$$

with appropriate initial data and boundary conditions, and where the parameter $a > 0$ jumps (it is a discontinuous function of x) at an internal interface.

The proper upwind numerical flux condition is not obtained by using the internal Lax-Friedrichs flux. It can be shown that this is obtained by the numerical flux

$$\begin{aligned} n\cdot\Gamma^* &= \frac{1}{a^l + a^r}(a^l n\cdot\Gamma^r + a^r n\cdot\Gamma^l + a^l a^r(u^l - u^r)) \\ &= \frac{a^r a^l}{a^l + a^r} n((u^r + u^l) + (nu^l - nu^r)) \end{aligned}$$

where ^{r} and ^{l} denote the right and left side of the interface respectively.

Since the down side coincides with the left side in 1D (*n=1*), this condition can be expressed by setting

```
g=-2*down(a)*up(a)*down(u)/(down(a)+up(a)).
```

In general and in higher dimensions, one typically needs the down (and up) versions of the mesh normal to express these conditions. For example, when the sign of the normal is unknown (i.e which side of an interface COMSOL Multiphysics considers the up/down side) the above condition can be entered as

```
g = -down(a)*up(a)/(down(a)+up(a))*dnx*(up(u)+down(u)+dnx*(down(u)-up(u)))
```

Here `dnx` means the down side normal (`dnx=-unx`).

It is important to use the mesh version of this vector in higher dimensions on curved boundaries. For example `dnxmesh`, `dnymesh` and `dnzmesh` denotes the x, y, and z components of mesh normal vector on the down side.

BOUNDARY SELECTION

In the **Selection** list, add the boundaries to define as a flux. Only interior boundaries can be selected.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

About Auxiliary Equation-Based Nodes

The PDE Interfaces provide an environment for specifying a model completely in terms of equations. In many cases, however, you may only be interested in adding an equation term or a constraint to the PDE implemented by a physics interface.

The auxiliary equation-based nodes are found under the context submenus **More>**, **Edges>**, and **Points>**. To display these in the context menu, click the **Show** button ( ) on the **Model Builder** toolbar and select **Advanced Physics Options**.

These are the available nodes (listed in alphabetical order):

- Auxiliary Dependent Variable
- Discretization (Node)
- Pointwise Constraint
- Weak Constraint
- Weak Contribution (PDEs)
- Weak Contributions on Mesh Boundaries

- The PDE Interfaces
- Domain, Boundary, Pair, Edge, and Point Conditions for PDEs

Weak Contribution (PDEs)

The **Weak Contribution** node is available in all interfaces and for all geometry levels, including the global level. The node adds an arbitrary contribution to the total system of equations. Its weak form expression may contain the `test()` operator acting on any dependent variable in the model and therefore add contributions to any equation.

To create an independent weak form equation rather than a weak form contribution, add extra states (dependent variables), right-click the **Weak Contribution** node and add an Auxiliary Dependent Variable. You can then use the auxiliary dependent variables in the weak-form expression.

To add this node, click the **Show** button ( ) and select **Advanced Physics Options**. Then, in the **Model Builder**, right-click the main interface node and, depending on the geometric entity level, select **More>Weak Contribution**, **Edges>Weak Contribution**, **Points>Weak Contribution**, or **Global>Weak Contribution**.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

In the **Selection** list, add the geometric entities (domains, boundaries, edges, or points) where you want to add a weak-form contribution.

WEAK CONTRIBUTION

Enter the weak-form contribution in the **Weak expression** field. For example, `-ux*test(ux)-uy*test(uy)+1*test(u)` is the weak formulation of a Poisson's equation with u as the dependent variable, and 1 as the source term on the right-hand side. To add a time derivative as in the time-dependent coefficient form PDE, add `-ut*test(u)` (notice the sign and the syntax `ut` for the time derivative of u).

QUADRATURE SETTINGS

These settings affect the numerical integration, and you do normally not need to change them. The **Use automatic quadrature settings** check box is selected by default, meaning that the settings are taken from the main equation in the interface.

If the check box is cleared, the following settings become available:

Integration Order

The **Integration order** specifies the desired accuracy of integration during discretization. Polynomials of at most the given integration order are integrated without systematic errors. For smooth constraints, a sufficient integration order is typically twice the order of the shape function. For example, the default integration order for second-order Lagrange elements is 4. The integration order is a positive integer.

Integrate on Frame

The **Integrate on frame** setting determines which frame to base the integration on—**Spatial**, **Material**, **Mesh**, or **Geometry**. The default frame is the one used for the physics interface.

Multiplication by 2πr

This section is available for 2D axisymmetric and 1D axisymmetric models

The option effectively controls the meaning of the numerical flux and natural boundary condition. By default, the **Multiply by 2πr** check box is selected, which is consistent with the implementation in the physics interfaces. This convention defines all fluxes as per unit area and the natural boundary condition per length and full revolution. Click to clear the check box to work with fluxes and natural boundary conditions per length and radian.

Weak Contribution (ODEs and DAEs)

Weak Contributions on Mesh Boundaries

The **Weak Contribution on Mesh Boundaries** node is available on the domain level in all physics interfaces. To add this node, click the **Show** button ( ) and select **Advanced Physics Options.** Then right-click the main PDE node and select **More>Weak Contribution on Mesh Boundaries** at the domain level.

This node is very similar to the Weak Contribution (PDEs) node on the boundary level, except that it requires a domain selection. Otherwise, the settings are identical. The domain selection is interpreted as all mesh element boundaries (edges in 2D and faces in 3D) in the interior of the selected domains.

Element edges or faces that are part of a real boundary are not included in this selection.

Use a **Weak Contribution on Mesh Boundaries** node to set up boundary conditions between every pair of adjacent mesh elements in its domain selection. To access values in the two adjacent elements, use the `up()` and `down()` operators. In practice, this node must be used together with discontinuous shape functions to implement a discontinuous Galerkin method. You can also right-click to add an Auxiliary Dependent Variable subnode.

up and down (operator)

Auxiliary Dependent Variable

If a complete equation is specified on weak form in a Weak Contribution (PDEs) or Weak Contributions on Mesh Boundaries node and a new dependent variable that is not part of the physics interface is used, right-click either of these nodes to add an **Auxiliary Dependent Variable** subnode.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node; that is, the selection is automatically selected and is the same as for the interface. You can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

In rare cases, more than one **Auxiliary Dependent Variable** node can be used on subsets of the **Weak Contribution** node's selection (to use different initial values, for example).

AUXILIARY DEPENDENT VARIABLE

Define the name and properties of the auxiliary dependent variable. Enter the:

- **Field variable name** (the dependent variable).
- **Initial value** for the dependent variable. The default is 0.
- **Initial time derivative** for the dependent variable. The default is 0.

Under **Discretization**:

- Except for the **Compute boundary fluxes** check box (which is not available), the rest of the settings are the same as for The Coefficient Form PDE Interfaces.
- The value type of auxiliary dependent variables is always **Complex** when **Split complex variables in real and imaginary parts** is activated in the Compile Equations node of any solver sequence used.

Discretization Section Shape Function Types and Element Orders

About Explicit Constraint Reaction Terms

In the Weak Constraint and Pointwise Constraint nodes you specify an expression, R, which is forced equal to zero. Optionally, you may also specify how reaction terms are applied. The default setting, **All physics (symmetric)**, applies reaction terms based on the constraint expression in a way that preserves the symmetry of a symmetric unconstrained system of equations. These reaction terms are uniquely defined by the symmetry requirement.

Selecting **User defined** from the **Apply reaction terms on** list lets you specify the constraint reaction terms explicitly, using a syntax borrowed from weak form modeling.

- Constraint Reaction Terms
- About Weak Form Modeling

CONSTRAINT REACTION TERM EXAMPLE

For example, in a three-variable problem for variables u_1, u_2 and u_3, specify the constraints (using the **Constraint expression** field in two separate Weak Constraint or Pointwise Constraint nodes)

$$\begin{cases} 0 = R_1(u_1, u_2) = 2u_1 - 3u_2 & \text{on } \partial\Omega \\ 0 = R_2(u_2, u_3) = 2u_2 + 3u_3 & \text{on } \partial\Omega \end{cases}$$

Note that both constraints involve both more than one variables and that to each constraint corresponds a Lagrange multiplier variable, μ_1 and μ_2, respectively. The weak equation corresponding to these constraints is

$$\int_{\partial\Omega} \hat{\mu}_1 R_1 + \hat{\mu}_2 R_2 \;=\; \int_{\partial\Omega} \hat{\mu}_1(2u_1 - 3u_2) + \hat{\mu}_2(2u_2 + 3u_3) \;=\; 0$$

where μ_1 and μ_2 are the test functions corresponding to the Lagrange multipliers.

This integral equation must be respected for every possible value of the Lagrange multiplier test functions. The only difference between a weak constraint and a pointwise constraint in this respect is that the Lagrange multiplier test functions in a weak constraint are nonzero over the elements adjacent to each mesh node, while the pointwise constraint test functions are nonzero only at the nodes. Therefore, weak constraints are enforced in a local weighted average sense, while pointwise constraints are enforced exactly at the mesh nodes. For this discussion, the difference is not important.

The default reaction terms, applied symmetrically to all dependent variables in the model, are defined simply by switching the test operator (here denoted by the circumflex “^”) from the Lagrange multipliers to the constraint expressions. Since the test operator is a linear differential operator, the weak form symmetric reaction term is

$$\int_{\partial\Omega} \mu_1 \hat{R}_1 + \mu_2 \hat{R}_2 \;=\; \int_{\partial\Omega} \mu_1\left(\frac{\partial R_1}{\partial u_1}\hat{u}_1 + \frac{\partial R_1}{\partial u_2}\hat{u}_2\right) + \mu_2\left(\frac{\partial R_2}{\partial u_2}\hat{u}_2 + \frac{\partial R_2}{\partial u_3}\hat{u}_3\right) \;=$$

$$\int_{\partial\Omega} \mu_1(2\hat{u}_1 - 3\hat{u}_2) + \mu_2(2\hat{u}_2 + 3\hat{u}_3)$$

The user-defined **Constraint force expression** to enter in a Weak Constraint or Pointwise Constraint node to explicitly recreate these symmetric reaction terms may be identified as the expressions multiplying the Lagrange multipliers. The test operator is denoted `test()` in weak expression syntax. Therefore, the constraint force expression for constraint $R_1 = 2u_1 – 3u_2 = 0$ is `test(2*u1-3*u2)` or, equivalently, `2*test(u1)-3*test(u2)`. The corresponding expression for R_2 is `test(2*u2+3*u3)` or, after linearization, `2*test(u2)+3*test(u3)`.

As a general rule, anything that multiplies `test(u1)` appears as a source term in the u_1 equation. Similarly, coefficients of `test(u2)` and `test(u3)` are source terms in the u_2 and u_3 equations, respectively. The symmetric reaction terms from the R_1 constraint contains both a `test(u1)` and a `test(u2)` term. Therefore, its generalized reaction force affects both these equations, while the reaction terms from R_2 affect the u_2 and u_3 equations.

Now suppose that u_1 and u_2 in reality represent components of the same vector, $\mathbf{u}$, while u_3 can be seen as en external quantity which should affect, but not be affected by, the value of $\mathbf{u}$. The symmetric reaction term from constraint R_2 violates this assumption and must be modified. In a user-defined **Constraint force expression**, write for example just `test(u2)` to apply reaction terms only as a generalized reaction force in the u_2 equation and leave the u_3 equation unaffected. This corresponds to a weak form integral contribution

$$\int_{\partial\Omega} \mu_1(2\hat{u}_1 - 3\hat{u}_2) + \mu_2\hat{u}_2$$

Nonsymmetric reaction terms do not always lead to a well-posed problem. If, in the example above, you would set the **Constraint force expression** also for the first constraint to `test(u2)`, the Lagrange multipliers become nonunique because only their sum enters the u_2 equation. At the same time there is no means to enforce any constraint on u_1 because that equation contains no Lagrange multiplier at all.

Pointwise Constraint

To add a **Pointwise Constraint** node on the domain, boundary, edge, or point level, click the **Show** button ( ) and select **Advanced Physics Options.** Then, depending on the geometric entity level, select **More>Pointwise Constraint** at the domain or boundary level, **Edges>Pointwise Constraint**, or **Points>Pointwise Constraint** from the context menu. There is no global pointwise constraint option.

This node adds standard pointwise constraints, similar to the ones used by boundary conditions of a constraint type in the physics interfaces.

Use **Pointwise Constraint** nodes to add extra constraints to a physics interface and to assume complete control over constraint reaction terms and points of application.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entities (domains, boundaries, edges, or points) to which you want to add a pointwise constraint. The default is **Manual**, but you can also choose, depending on the geometric entity level, **All domains**, **All boundaries**, **All edges**, or **All points** in the model.

POINTWISE CONSTRAINT

These settings are similar to those for the Weak Constraint, but pointwise constraints do not need explicit, named, Lagrange multipliers. Instead, implicit Lagrange multipliers are eliminated by the solvers, together with the degrees of freedom being constrained.

Select an option from the **Apply reaction terms on** list—**All physics (symmetric)** (the default) or **User defined**. For either option, enter a **Constraint expression**, which COMSOL Multiphysics constrains to 0. For example, entering `2-(u+v)` constrains $u+v$ to the value 2.

If **User defined** is selected, enter also a **Constraint force expression**. Note that the constraint force expression must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the *u* equation with reaction terms.

See Coefficient Form PDE for all the settings and Compact and Standard Notations for Classical PDEs for the equations that the Classical PDE interface solve.

- About Explicit Constraint Reaction Terms
- Constraint Reaction Terms

DISCRETIZATION

Select a **Shape function type** (finite element types)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (the order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

In most cases, use the same shape function type and order for the pointwise constraint as for the dependent variables being constrained. If dependent variables of different order appear in the constraint expression, select the highest order for the pointwise constraint. Notable exceptions are the Hermite and Argyris shape functions, which should be constrained by Lagrange elements of the corresponding order.

The **Frame** list is available when there is more than one unique frame in the model. In this case, select **Geometry**, **Mesh**, **Spatial** (the default), or **Material** from the **Frame** list. This choice only affects how COMSOL computes derivatives of Lagrange multipliers and in general does not make any difference at all.

Weak Constraint

To add a **Weak Constraint** node on a domain, boundary, edge, or point level, click the **Show** button ( ) and select **Advanced Physics Options**. Then, depending on the geometric entity level, select **More>Weak Constraint**, **Edges>Weak Constraint**, or **Points>Weak Constraint** from the context menu. There is no global weak constraint option.

The **Weak Constraint** node adds an extra dependent variable, known as a Lagrange multiplier, and a weak equation, which together enforce the specified constraint.

If the weak constraint is redundant in the sense that some other weak or pointwise constraint also controls the value of the constraint expression, the Lagrange multiplier becomes under-determined. This makes the discrete system of equations singular and therefore virtually impossible to solve, even if the solution for the main dependent variables is unique.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node; that is, the selection is automatically selected and is the same as for the interface. You can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

WEAK CONSTRAINT

Select an option from the **Apply reaction terms on** list—**All physics (symmetric)** (the default) or **User defined**. For either option, enter a **Constraint expression**, which COMSOL Multiphysics constrains to 0. For example, entering `2-(u+v)` constrains $u+v$ to the value 2.

If **User defined** is selected, enter also a **Constraint force expression**. The constraint force expression must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the u equation with reaction terms.

- About Explicit Constraint Reaction Terms
- Constraint Reaction Terms

QUADRATURE SETTINGS

These settings affect the numerical integration, and you do normally not need to change them. The **Use automatic quadrature settings** check box is selected by default, meaning that the settings are taken from the main equation in the interface. If the check box is cleared, the following settings become available:

Integration Order

The **Integration order** is a positive integer that specifies the desired accuracy of integration during discretization. Polynomials of at most the given integration order are integrated without systematic errors. For smooth constraints, a sufficient integration order is typically twice the order of the shape function. For example, the default integration order for second-order Lagrange elements is 4.

When constraints may be discontinuous at points inside the mesh elements, raising the integration order may improve accuracy.

Integrate on Frame

The **Integrate on frame** setting determines which frame to base the integration on—**Spatial**, **Material**, **Mesh**, or **Geometry**. The default frame is the one used for the physics interface.

Multiplication by $2\pi r$

This section is available for 2D axisymmetric and 1D axisymmetric models

By default, the **Multiply by $2\pi r$** check box is selected to make the Lagrange multiplier represent, for example, the flux per area rather than by length and full revolution. If the check box is cleared, the Lagrange multiplier is not multiplied by $2\pi r$ where it is used in the constraint equation, and therefore represents flux per length and full revolution.

When weak constraints are activated under **Constraint Settings** in other constraint-type boundary conditions, there is no automatic multiplication by $2\pi r$ for axial symmetry.

LAGRANGE MULTIPLIER

Enter a **Lagrange multiplier variable** (the default name is `lm`) and an **Initial value**. Change the name if required, for example because a variable name conflicts.

DISCRETIZATION

Select a **Shape function type** (finite element types)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (the order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

Always use the same shape function type for the weak constraint as for the variables that are constrained, possibly with lower-order elements for the weak constraint. In some cases (for example, when constraining derivatives) the system of equations can become singular. The reason is usually that there are redundant Lagrange multiplier degrees of freedom in the model. Try to lower the order of the Lagrange multiplier variables or use constraints on the Lagrange multiplier to remove some degrees of freedom.

The **Frame** list is available when there is more than one unique frame in the model. In this case, select **Spatial** or **Material** from the **Frame** list. This affects only how derivatives of Lagrange multipliers are computed. These are normally not used in the constraint equations, but may be of interest for postprocessing.

The value type of auxiliary dependent variables is always **Complex** when **Split complex variables in real and imaginary parts** is activated in the Compile Equations node of any solver sequence used.

Discretization Section Shape Function Types and Element Orders

Discretization (Node)

To display the option, click the **Show** button (👁) and select **Advanced Physics Options** then right-click the main physics interface node and select **Global>Discretization** to add extra **Discretization** nodes.

Use the **Discretization** nodes to specify multiple discretizations for the same physics, typically using different element orders. In study steps and the **Multigrid Level** subnodes, you select among the discretizations specified for each physics interface. This may be used to manually create a multigrid hierarchy, just to compare the effect of different discretizations in different studies.

The **Discretization** settings window usually has one section and it is the same as the **Discretization** section found in the main physics interface settings window.

- Additional Advanced Physics Options
- Discretization Section Shape Function Types and Element Orders
- Advanced Physics Sections
- For information about how to specify the splitting of complex variables, see Compile Equations.

Modeling with ODEs and DAEs

The ODE and DAE Interfaces

Adding ODEs, DAEs, and Other Global Equations

When working on complex models, single named degrees of freedom, or *states*, may be needed to track and update information that is not logically related to any particular point in space. The evolution of these states is generally governed by equations, which are independent of space, but often time-dependent. In particular, such situations arise when modeling physics in interaction with an external system, for example, a controller or an electrical circuit built from standard components. It is often possible to describe such external systems by a system of ordinary differential equations (ODEs) with a limited number of degrees of freedom.

The Global ODEs and DAEs Interface has a Global Equations node that is designed for implementing this type of external equation. Such equations are often tightly coupled to a model in a physical domain. The Global Equations node is also available for any of the physics interfaces.

To access the node, right-click the main interface in the **Model Builder** and select **Global>Global Equations**.

Use the Global Equations node for ODEs, differential algebraic equations, purely algebraic equations and conditions, and transcendental equations, or to add degrees of freedom to a model using the introduced states. Possible uses include:

- Controllers
- Rigid-body mechanics
- Nonlinear eigenvalue problems
- Continuation
- Integral constraints
- Augmented or generalized equations

PRESENTING RESULTS FOR GLOBAL EQUATIONS

The state variables are scalar values and they are available globally. To view the results for an ODE, use the Line Graph, Point Graph, and Global plot types, and Global Evaluation for displaying the numerical solution.

- Plot Groups and Plots
- Derived Values and Tables

Solving ODEs: An Example

As an example of ODEs, the following equations are the *Lotka-Volterra equations*

$$\dot{r} = ar - brf$$
$$\dot{f} = -cf + drf$$

where r is the rabbit population, and f is the population of foxes. This is an example of a system of two coupled ODEs.

Enter these equations as `a*r-b*r*f-rt` and `-c*f+d*r*f-ft`, where a, b, c, and d are scalar values defined using the Parameters node. For this first-order ODE, specify initial values for r and f.

About Parameters, Variables, and Expressions

Solving Algebraic and Transcendental Equations: An Example

As an example of an algebraic equation, consider the equation $f(u) = 0$ for

$$f(u) = u^3 + u - 2$$

This equation has a single real-valued root at $u = 1$.

1 In the **Model Builder**, click the Global Equations node in a Global ODEs and DAEs interface.

2 In the **Global Equations** settings window, enter `u` in the **Name** column and `u^3+u-2` in the **f(u,ut,utt,t)** column (both entries on the same row).

3 Solve this using a stationary solver.

4 In the **State variable u** (**Global Evaluation**) settings window, click the **Evaluate** button (=), and the value of the root displays in the **Table** window.

As an example of a transcendental equation, consider the equation $f(u) = 0$ for

$$f(u) = e^{-u} - u$$

A root to this equation is approximately 0.56714. To enter it for the node:

1 In the **Model Builder**, click the Global Equations node.

2 Enter `u` in the **Name** column and `exp(-u)-u` in the **f(u,ut,utt,t)** column (both entries on the same row).

3 Compute the solution.

4 In the **State variable u** (**Global Evaluation**) settings window, click the **Evaluate** button (=), and the value of the root (rounded to 0.56714) displays in the **Table** window.

Derived Values and Tables

Distributed ODEs and DAEs

For ODEs and DAEs in domains, on boundaries and edges, and at points, The Distributed ODEs and DAEs Interfaces are available to solve the following ODE (or DAE):

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} = f$$

Because it is defined on a geometry but is space independent, it solves the ODE or DAE as a distributed ODE or DAE, with a solution that is defined as a field but with no space variation.

The ODE and DAE Interfaces

The Global ODEs and DAEs Interface

The **Global ODEs and DAEs** interface ($\frac{d}{dt}$), found under the **Mathematics** branch (Δu) when adding an interface, is used to add global space-independent equations that can represent additional states. The equations can be ODEs, algebraic equations, and DAEs.

The main default node is **Global Equations**, used to define the global equations including the names of the variables (states), the required initial values, and an optional description. Then, from the **Physics** toolbar, add other nodes, for example, Global Equations, Global Constraint, or Weak Contribution (ODEs and DAEs). You can also right-click **Global Equations and ODEs** to select physics from the context menu.

- Solving Algebraic and Transcendental Equations: An Example
- Adding ODEs, DAEs, and Other Global Equations

About ODEs, Initial-Value Problems, and Boundary-Value Problems

ODEs (ordinary differential equations) are often divided into these two types:

- *Initial-value problems* (IVPs), where the solution u and its derivatives (often with respect to time) are specified in one point (in time) so that $u(0)$ and $u'(0)$ are known, so the system is assumed to start at a fixed initial point. The Global Equations node in COMSOL Multiphysics supports such IVPs described with an equation in the following form $f(u, \dot{u}, \ddot{u}, t) = 0$ including initial values for u and its derivatives.
- B*oundary-value problems* (BVPs), where the solution u have specified values at a pair of points such as $u(0)$ and $u(1)$; that is, the points 0 and 1 are regarded as boundary points of the domain for the problem. In COMSOL you can specify such a BVP as a stationary 1D PDE, where the 1D interval represents the independent variable and the u is the dependent variable in the interval.

Global Equations

A default **Global Equations** node is added to The Global ODEs and DAEs Interface. Right-click to add additional **Global Equations**, Weak Contribution (ODEs and DAEs), and Global Constraint nodes.

In any other physics interface, click the **Show** button ( ) and select **Advanced Physics Options**. Then right-click the physics interface and select **Global>Global Equations** to add a node directly, without needing to add a separate Global ODEs and DAEs interface.

GLOBAL EQUATIONS

The global equations that you can solve have the following form:

$$f(u, \dot{u}, \ddot{u}, t) = 0$$

with the initial conditions $u(t_0) = u_0$ and $u_t(t_0) = u_{t,0}$ (where the subscript t indicates the time derivative). Several equations can be added and the equation can be coupled.

The first time derivative of u is written `ut`, and the second time derivative of u is `utt`. With time derivatives, this equation is an ODE (ordinary differential equation). With no time derivatives, the equation is an algebraic equation or a transcendental equation. If some equations include time derivatives and others not, the system is a DAE (differential-algebraic equation).

> **!** Initial conditions are necessary for ODEs and DAEs. For the DAEs, it is important to specify a set of initial conditions consistent with the algebraic part of the system. Otherwise, the solvers may be forced to modify the initial values to make them consistent, or they may fail completely.

In the **Global Equations** table, each row corresponds to a named state; that is, it defines a single degree of freedom and one equation.

> The selected row in the table of global equations may also be edited using the **Name**, **f(u,ut,utt,t)**, **Initial value (u_0)**, **Initial value (u_t0)**, and **Description** fields underneath the table.

In each column enter as follows.

- Enter the **Name** of the state variable. This also defines time-derivative variables. If a state variable is called `u`, its first and second time derivatives are `ut` and `utt`, respectively. These variables become available in all geometries. Therefore the names must be unique.
- Use the **f(u,ut,utt,t)** column to specify the right-hand side of the equation that is to be set equal to zero.

  The software then adds this global equation to the system of equations. When solving the model, the value of the state variable `u` is adapted in such a way that the associated global equation is satisfied. All state variables and their time derivatives can be used as well as any parameters, global variables, and coupling operators with a scalar output and global domain of definition in the **f(u,ut,utt,t)** column. The variables can be functions of the state variables in the global equations. Setting an equation for a state is optional. The default value of `0` means that the software does not add any additional condition to the model.
- If the time derivative of a state variable appears somewhere in the model during a time-dependent solution, the state variable needs an initial condition. Models that contain second time derivatives also require an initial value for the first time derivatives of the state variables. Set these conditions in the third (**Initial value (u_0)**) and fourth (**Initial value (u_t0)**) columns.
- Enter comments about the state or the equation in the last column, **Description**.
- To add another equation, make additional entries in first empty row.

  Move equation rows up and down using the **Move Up** (↑) and **Move Down** (↓) buttons. To remove an equation, select some part of that equation's row in the table and click the **Delete** button ( ).

  Save the definitions of the global equations to a text file by clicking the **Save to file** button ( ) and using the **Save to File** dialog box that appears. To load a text file with global equation definitions, use the **Load from File** button ( ) and using the **Load from File** dialog box that appears. Data must be separated by spaces or tabs (or be in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®).

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**.

The **Discretization** section for **Global Equations** is used to specify the **Value type** (**Real** or **Complex**) of the variables the **Split complex variables in real and imaginary parts** setting is activated in the **Compile Equations** node of any solver sequence. The default for split complex variables setting is to be not active and in that case you do not need to specify the value type for global equations variables (the value type specified would be ignored in such case). The value type for all the variables defined by this **Global Equations** node is selected in the **Value type when using splitting of complex variables** selection. The default value type is **Complex**.

Global Constraint

Right-click The Global ODEs and DAEs Interface node and select **Global Constraint** to add a global (space-independent) constraint. With such a constraint you can, for example, make the sum of some global variables equal to a fixed number.

In any other physics interface, click the **Show** button ( ) and select **Advanced Physics Options**. Then right-click the physics interface and select **Global>Global Constraint**.

GLOBAL CONSTRAINT

Choose an option from the **Apply reaction terms on** list—**All physics (symmetric)** (the default) or select **User defined** to define reaction terms. For either choice, enter a **Constraint expression**. The default is 0. For example, entering `2-(u+v)` constrains $u+v$ to the value 2. If **User defined** is selected, enter a **Constraint force expression**.

The **Constraint force expression** must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the u equation with reaction terms. See Constraint Reaction Terms and Pointwise Constraint for more information.

Weak Contribution (ODEs and DAEs)

Another option is to enter equations in the weak form using the **Weak Contribution** node. This can be convenient in advanced modeling because it controls the test variables multiplying the equations. Wherever a test function of a state variable appears (in the Global Equations node or elsewhere in the model), whatever it multiplies ends up in the same equation in the discrete system

There can be zero or more weak expressions, regardless of the number of states.

WEAK CONTRIBUTION

Enter the expression that contains the weak formulation in the **Weak expression** field.

See The Wall Distance Interface for an example of how to write an ODE using a weak formulation.

The Distributed ODEs and DAEs Interfaces

The interfaces for distributed ODEs and DAEs ($\frac{d}{dt}$)—**Domain ODEs and DAEs (dode)**, **Boundary ODEs and DAEs (bode)**, **Edge ODEs and DAEs (eode)**, and **Point ODEs and DAEs (pode)**—are found under the **Mathematics>ODE and DAE Interfaces** branch ($\frac{d}{dt}$) when adding an interface. These interfaces provide the possibility to solve distributed ODEs and DAE in domains, on boundaries and edges, and at points.

When any of these interfaces are added, these default nodes are also added to the **Model Builder**—**Distributed ODE** and **Initial Values**. Then, from the **Physics** toolbar, add other nodes that implement, for example, other algebraic equations. You can also right-click the node to select physics from the context menu.

The following sections provides information about nodes available in the distributed ODEs and DAEs interfaces.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `dode` (in domains), `bode` (on boundaries), `eode` (on edges), or `pode` (at points).

Under **Discretization**, and except for the **Compute boundary fluxes** check box (which is not available), the rest of the settings are the same as for The Coefficient Form PDE Interfaces.

- Discretization Section Shape Function Types and Element Orders
- Distributed ODE
- Initial Values (the same as for the PDE interfaces)

Distributed ODE

This is the default equation for a **Distributed ODE** in an interface for distributed ODEs and DAEs. Specify the coefficients for an ODE with the following equation coefficients:

$$e_{\mathrm{a}}\frac{\partial^2 u}{\partial t^2}+d_{\mathrm{a}}\frac{\partial u}{\partial t} = f$$

- e_{a} is the *mass coefficient*.
- d_{a} is a *damping coefficient* or a *mass coefficient*.
- f is the *source term*.

The settings are the same as for the Coefficient Form PDE.

Algebraic Equation

This node adds an **Algebraic Equation** in an interface for distributed ODEs and DAEs. Specify an algebraic equation as

$$f = 0 \tag{16-14}$$

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

The default setting is to include **All domains**, **All boundaries**, **All edges**, or **All points** in the model. To choose specific geometric entities, select **Manual** from the **Selection** list.

ALGEBRAIC EQUATION

Enter an expression f for the equation $f = 0$ that defines the algebraic equation in the **f** field. If there are multiple dependent variables, there is a vector of algebraic equations for each variable.

The Events Interface

The **Events** interface ( ) is used to create solver events. An event can be explicit or implicit, and the difference is that for explicit events you must specify the exact time when the event occurs. When an event occurs, the solver stops and provides a possibility to reinitialize the values of states and dependent variables.

From the **Physics** toolbar, **Global>** submenu, add nodes, for example, Discrete States, Indicator States, Explicit Event, or Implicit Event. You can also right-click **Events** to select physics from the context menu.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `ev`.

> If you have the Batteries & Fuel Cells Module, see Capacity Fade in a Lithium-Ion Battery for the use of this interface and the Discrete States, Indicator States, and Implicit Event nodes (model library path **Batteries_and_Fuel_Cells_Module/Batteries/capacity_fade**).

Discrete States

Use the **Discrete States** node ( ) to define discrete state variables that do not have to be continuous in time. They are often used as logical help variables that control what expression to use in a weak expression or constraint.

DISCRETE STATES

Enter values or expressions in the table for the **Name**, **Initial value (u0)**, and enter a **Description**.

Indicator States

Use the **Indicator States** node ( ) to define state variables that the solver uses to trigger implicit events, which occurs at a zero crossing of an indicator state variable.

> This behavior implies, for example, that an event of the type $z > 0$, where z is an indicator state, can be detected accurately because the zero crossing of z is found. But if the condition (in Event Conditions) is formulated as, for example, $z > 1$, the event is not triggered accurately.

INDICATOR VARIABLES

Multiple state variable can be defined in the table together with corresponding g-coefficients and initial values. A state variable, u, gets its value by solving the following equation:

$$\mathrm{nojac}(g(v, \dot{v}, \ddot{v}, t)) - u = 0$$

with the initial conditions $u(t_0) = u_0$ (where the dot notation indicates time derivatives). You use the state variables in the implicit event condition that you specify for the Implicit Event node.

Enter values or expressions in the table for the **Name**, g-coefficients **g(v,vt,vtt,t)**, **Initial value (u0)**, and (optionally) enter a **Description**.

Explicit Event

An **Explicit Event** node () specifies an event that occurs at predefined times entered in the **Event Timings** section. When an event occurs, it is possible to specify reinitialization of global states in the **Reinitialization** section. Right-click an **Explicit Event** node to add the Reinitialization on Domains, Boundaries, Edges, or Points node.

EVENT TIMINGS

Write the first time an explicit event occur in the **Start of event** edit field. It is also possible to repeatedly trigger the event by entering the period in the **Period of event** field.

REINITIALIZATION

In the table, enter information in the **Variable** and **Expression** columns for the global state variables that the event reinitializes to the given expression.

Implicit Event

Use the **Implicit Event** node () to specify an event that occurs when an condition involving an indicator state is fulfilled. When an event occurs, it is possible to specify reinitialization of global state variables. Right-click an **Implicit Event** node to add a Reinitialization on Domains, Boundaries, Edges, or Points node.

EVENT CONDITIONS

Enter the condition in the **Condition** field. When the condition changes its state from false to true, this triggers the implicit event.

REINITIALIZATION

In the table, enter information in the **Variable** and **Expression** columns for the global state variables that the event reinitializes to the given expression.

Reinitialization on Domains, Boundaries, Edges, or Points

The **Reinitialization on Domains**, **Reinitialization on Boundaries**, **Reinitialization on Edges**, or **Reinitialization on Points** node can reinitialize dependent variables defined on domains when an event occurs. Right-click an Explicit Event or Implicit Event node to add this node.

REINITIALIZATION

In the table under **Variables** enter the dependent variables. In the **Expression** column, enter the corresponding information for each variable as defined on domains, boundaries, edges, or points that the event reinitializes.

The Wall Distance Interface

The **Wall Distance** interface ( ), found under the **Mathematics** branch (Δu) when adding an interface, has the equations and boundary conditions for calculating the distance to walls in fluid-flow simulation using a modified *eikonal equation*, solving for a dependent variable G that is related to the wall distance. The main node is the **Distance Equation** node, which adds the distance equation (modified Eikonal equation) and provides an interface for defining the reference length scale.

When this interface is added, these default nodes are also added to the **Model Builder**—**Wall Distance**, **Distance Equation**, and **Initial Values**. Default boundary conditions are also added—a homogeneous Neumann condition that does not appear in the Model Builder. Right-click the **Wall Distance** node to add boundary conditions for walls.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `wd`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

Enter a **Smoothing parameter** σ_w. in Equation 16-17. The default value is 0.1.

DEPENDENT VARIABLES

The dependent variable is the **Reciprocal wall distance** G. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another reciprocal distance field in the model, the interfaces shares degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**. Select **Quadratic** (the default), **Linear, Cubic**, or **Quartic** for the **Reciprocal wall distance**.

- Show More Physics Options
- Domain and Boundary Nodes for the Wall Distance Interface and Initial Values

Domain and Boundary Nodes for the Wall Distance Interface

The Wall Distance Interface has these domain settings and boundary conditions available:

- Distance Equation

- Initial Values
- Wall

The default boundary condition, a homogeneous Neumann condition, applies to all boundaries that are not walls:

$$\nabla G \cdot \mathbf{n} = 0$$

This node does not display in the model tree.

For axisymmetric models, COMSOL Multiphysics automatically takes the axial symmetry boundaries (at $r = 0$) into account, and prohibits the wall boundary node from being defined here.

Distance Equation

The **Distance Equation** node adds to Equation 16-17, and the **Distance Equation** form contains the following sections for defining the length scale.

DOMAIN SELECTION

In the **Selection** list, add the domains to use the distance equation with the length scale defined in this **Distance Equation** node.

DISTANCE EQUATION

Select an option for the **Reference length scale** l_{ref} (SI unit: m)—**Automatic** (the default) to use Equation 16-18 to compute l_{ref}, or **Manual** to enter a value in the field. The default value is 1 m.

Initial Values

The **Initial Values** node adds an initial value for the reciprocal wall distance that can serve as initial guess for a nonlinear solver.

DOMAIN SELECTION

In the **Selection** list, add the domains where you want to define the initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the **Reciprocal wall distance** G (SI unit: 1/m). The default value `wd.G0`.

Wall

The **Wall** node implements the following boundary condition for walls:

$$G = G_0 = \frac{2}{l_{\mathrm{ref}}}$$

BOUNDARY SELECTION

In the **Selection** list, add the boundaries to define as a wall.

Theory for Wall Distance

The Wall Distance Interface theory is described in this section:

The Eikonal Equation

Turbulence models often use the distance to the closest wall to approximate the mixing length or for regularization purposes. One way to compute the wall distance is to solve the Eikonal equation:

$$|\nabla D| = 1 \tag{16-15}$$

with $D = 0$ on solid walls and $\nabla D \cdot \mathbf{n} = 0$ on other boundaries. The Eikonal equation can be tricky to solve, and produces the exact distance to the closest wall. The modified equation described below is less computationally expensive to solve. It also uses a reference length to put more emphasis on solid objects larger than the reference length and reduce the emphasis on smaller objects. The introduction of a reference length produces a more relevant wall distance in the following instances:

- In convex regions of small dimensions, the wall distance is reduced to reflect the close proximity of several walls.
- When a small object, such as a thin wire for example, is present in free flow, the wall distance is affected only in a very small region around it.

Modified Eikonal Equation

COMSOL Multiphysics uses a modified Eikonal equation based on the approach in Ref. 1. This modification changes the dependent variable from D to $G = 1/D$. Equation 16-15 then transforms to

$$\nabla G \cdot \nabla G = G^4 \tag{16-16}$$

Additionally, the modification adds some diffusion and multiplies G^4 by a factor to compensate for the diffusion. The result is the following equation, which the Wall Distance interface uses:

$$\nabla G \cdot \nabla G + \sigma_{\mathrm{w}} G(\nabla \cdot \nabla G) = (1 + 2\sigma_{\mathrm{w}})G^4 \tag{16-17}$$

where σ_{w} is a small constant. If σ_{w} is less than 0.5, the maximum error falls of exponentially when σ_{w} tends to zero. The default value of 0.1 is a good choice for both linear and quadratic elements.

The boundary conditions for Equation 16-17 is $G = G_0 = C/l_{\mathrm{ref}}$ on solid walls and homogeneous Neumann conditions on other boundaries. The effect of C is that the solution becomes less smeared the higher the value of C. The error tends asymptotically to $0.2l_{\mathrm{ref}}$ as C tends to infinity, but making it very large destabilizes Equation 16-17. C is 2 in the Wall Distance interface.

The effect of l_{ref} is loosely speaking that the distance to objects larger than l_{ref} is represented accurately, while objects smaller than l_{ref} appears to be further away than their exact geometrical distance. For a channel, l_{ref} should typically be set to the channel width or there about. l_{ref} has a lower bound in that it must be larger than all cells adjacent to any boundary where the boundary condition $G = G_0$ is applied; otherwise, the solution displays oscillations. l_{ref} is the only parameter in the model, and the default value is half the shortest side of the geometry bounding box. If the geometry consists of several very slender entities, or if the geometry contains very fine details, this measure can be too large. Then define l_{ref} manually.

The initial value is by default defined as $G_0 = 2/l_{\mathrm{ref}}$, in correspondence with the boundary conditions.

The wall distance $D_{\mathrm{w}} = 1/G - 1/G_0$ is a predefined variable that is used for analysis. You have also access to a vector-valued variable that represents the direction toward the nearest wall, which is defined as

$$\frac{\nabla G}{\sqrt{\max(\nabla G^2, \mathrm{eps})}} \tag{16-18}$$

Reference for the Wall Distance Interface

1. E. Fares and W. Schröder, “A Differential Equation for Approximate Wall Distance,” *International Journal for Numerical Methods in Fluids*, vol. 39, pp. 743–762, 2002.

Curvilinear Coordinates

Introduction

A *curvilinear coordinate system* is a type of coordinate system where the coordinate lines can be curved. Using a curvilinear coordinate system can make it easier to, for example, define boundary conditions that follow a curved surface. The curvilinear coordinate systems in COMSOL Multiphysics define a vector field and an associated base vector system using four different methods:

- A diffusion method, which solves Laplace's equation $\Delta u = 0$ and computes the vector field as $-\nabla u$. Solve for the vector field using a Stationary study.
- An elasticity method, which computes the vector field from the solution with the eigenvalue closest to zero for the following eigenvalue problem:

  $$-\nabla\cdot(s(\nabla\cdot\mathbf{u})I+(\nabla\mathbf{u}+\nabla\mathbf{u}^T)) = \lambda\mathbf{u}$$

  Solve for the vector field using an Eigenvalue study.
- A flow method, which solves the following equation for $\mathbf{u}$ and p, and uses $\mathbf{u}$ as the vector field:

  $$\nabla\cdot[-pI+(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)] = 0$$
  $$\nabla\cdot\mathbf{u} = 0$$

  Solve for the $\mathbf{u}$ and p using a Stationary study.
- A user-defined vector field. No study step is required.

> Solve for the curvilinear coordinates (unless they are user-defined) in a separate study or separate study step, which you run before the study or study step where you solve for the physics that make use of the computed curvilinear coordinates. Simultaneously solving for the curvilinear coordinates and the physics that use them does normally not work.

The Curvilinear Coordinates Interface

With the **Curvilinear Coordinates** interface you can create a vector field $\mathbf{v}$ and a base vector system (with basis vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$) that can be used by other physics to specify, for example, external currents or anisotropic material properties of a bundle of wires or fibers. The resulting coordinate system can be—but does not have to be—curvilinear. Right-click the **Curvilinear Coordinates** node ( ) to add one of the available methods for computing the vector field for the curvilinear coordinates:

- Diffusion Method
- Elasticity Method
- Flow Method
- User Defined

All the nodes for these methods have an initially empty selection. You can use more than one vector field computation method (of the same or different types), each one in a different set of domains.

The **Curvilinear Coordinates** interface is available for 2D, 2D axisymmetric, and 3D geometries. The settings window contains the following sections:

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `cc`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

EQUATION

See Physics Nodes—Equation Section.

SETTINGS

The **Normalize vector field** check box is selected by default because a normalized vector field simplifies the use of curvilinear coordinates and a base vector system.

Select the **Create base vector system** check box to add a **Curvilinear System** node under **Definitions**, which is a Base Vector System node with a name that indicates that it is created by a **Curvilinear Coordinates** interface and contains the base vectors from the curvilinear coordinate computation. Selecting this check box also adds a Coordinate System Settings subnode for specifying the second basis vector $\mathbf{e}_2$ (the software then forms the third basis vector as the cross product of the first and second basis vectors).

Diffusion Method

Add a **Diffusion Method** node to compute the vector field based on Laplace's equation $\Delta u = 0$ with the vector field $\mathbf{v}$ defined as $\mathbf{v} = -\nabla u$ (divided by $|-\nabla u|$ if normalized). This method is a scalar "potential method" resulting in an incompressible vector field and is useful for geometries that are smooth but leads to concentrations at sharp corners. To define the equation in the geometry you can add the following boundary conditions:

- Wall (the default boundary condition)
- Inlet
- Jump
- Outlet

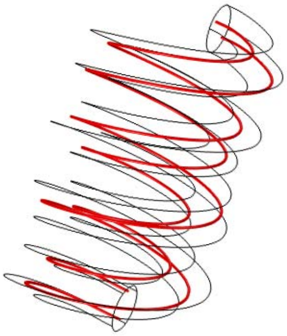


Figure 16-1: The computed velocity field in a helix using a Curvilinear Coordinates interface with a Diffusion Method.

Elasticity Method

Add an **Elasticity Method** node to compute the vector field based on the solution with the eigenvalue closest to zero for the following eigenvalue problem:

$$-\nabla\cdot(s(\nabla\cdot\mathbf{u})I+(\nabla\mathbf{u}+\nabla\mathbf{u}^T)) = \lambda\mathbf{u}$$

The vector field $\mathbf{v}$ is the same as $\mathbf{u}$ but normalized if normalization is selected.

The Elasticity method requires that you use an eigenvalue study.

This method is similar to the Coil Current Calculation approach used in the Multi-Turn Coil Domain feature in Magnetic Fields (in the AC/DC Module) and is useful for coil applications, for example. To define the equation in the geometry you can add the following boundary conditions:

- Wall (the default boundary condition)
- Outlet

In addition, a default **Inlet** node is added because one inlet must be added for the geometry to define the positive direction of the vector field for the curvilinear coordinate as indicated by the arrow that appears on the inlet boundary in the **Graphics** window.

Flow Method

Add a **Flow Method** node to solve the following equation for the vector $\mathbf{u}$ and the scalar p and use $\mathbf{u}$ as the vector field:

$$\nabla\cdot[-pI+(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)] = 0$$
$$\nabla\cdot\mathbf{u} = 0$$

The vector field $\mathbf{v}$ is the same as $\mathbf{u}$ but normalized if normalization is selected. This approach is equivalent to computing the flow of an incompressible fluid—that is, creeping flow or Stokes flow. The Flow method is useful for geometries with nonconstant cross sections. To define the equation in the geometry you can add the following boundary conditions:

- Wall (the default boundary condition)
- Inlet
- Jump
- Outlet

User Defined

Add a **User Defined** node to specify the vector field $\mathbf{u}$ as user-defined expressions for its components. In the **User Defined** section, enter the expressions for those components the text fields under **Vector field**. The vector field $\mathbf{v}$ is the same as $\mathbf{u}$ but normalized if normalization is selected. You can select any other coordinate system in the model from the **Coordinate system** list to use as the coordinate system for defining the vector field. The **Global coordinate system** is the default.

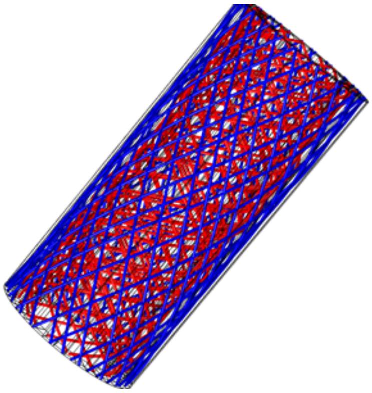


Figure 16-2: A model of arterial wall mechanics, using four user-defined Curvilinear Coordinates interfaces and a cylindrical coordinate system to define the wall pattern.

Inlet

Use an **Inlet** node to define where the vector field starts. Select the boundaries to define as the vector field's inlet in the **Boundary Selection** section. For the **Diffusion Method** and **Elasticity Method** it sets $\mathbf{u}$ to 1 at the inlet. For the **Flow Method** you can choose, in the **Inlet** section, to define the type of inlet using the **Type** list:

- Choose **Normal velocity** to specify the normal velocity $\mathbf{u}{\cdot}\mathbf{n}$ as a velocity u_n (default value: 1, SI unit: m/s) in the **Velocity** field.
- Choose **Velocity field** to specify the components of the velocity field $\mathbf{u}_\mathrm{in}$ in the text fields under **Velocity** (default: 0, SI unit: m/s). You can select any coordinate system for the Component from the **Coordinate system** list to use as the coordinate system for defining the velocity field. The **Global coordinate system** is the default.

Jump

Use a **Jump** node at a boundary to define a closed-loop vector field. You can add **Jump** nodes to interior boundaries. It is equivalent to a jump from 0 to 1 for the **Diffusion Method** and to a flow inlet and a flow outlet (with constant pressure) for the **Flow Method**; see Inlet above for the available settings. For the **Elasticity Method**, the jump condition is not applicable; instead, you can add an **Inlet** node to an interior boundary if needed.

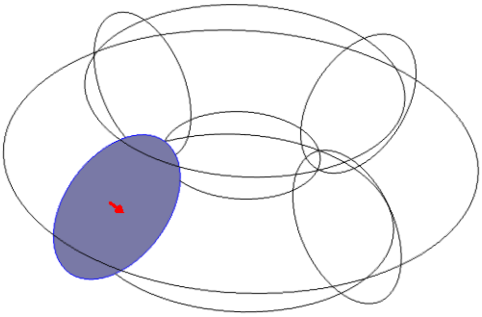


Figure 16-3: A Jump boundary condition applied to one of the interior boundaries of a torus, using a Diffusion method. The arrow indicates the direction of the vector field (the side where the value is 1).

Outlet

Use an **Outlet** node to define where the vector field ends. For the **Diffusion Method** this means that $u = 0$, for the **Elasticity Method** it sets $\mathbf{e} \times \mathbf{n}$ to 0. For the **Flow Method** it is a zero pressure and no stress condition: $p = 0$ and $(\nabla u + (\nabla u)^T) \cdot \mathbf{n} = 0$. Select the boundaries to define as the vector field's outlet in the **Boundary Selection** section.

Wall

The **Wall** node is the default boundary condition and defines the walls as boundaries where the normal component of the vector field is zero. It applies to all boundaries where you do not assign any of the other boundary conditions.

Coordinate System Settings

A **Coordinate System Settings** node is added by default if you have selected the **Create base vector system** check box in the **Curvilinear Coordinates** node's settings window. You use this node to specify the second basis vector for the created base vector coordinate system. The Curvilinear Coordinates interface solves for first basis vector and then computes the third basis vector for a full orthonormal coordinate system as the cross product of the first and second basis vector:

$$\mathbf{e}_1 = \frac{\mathbf{v}}{|\mathbf{v}|}$$

$$\mathbf{e}_2 = \frac{\mathbf{v}_2 - (\mathbf{v}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{|\mathbf{v}_2 - (\mathbf{v}_2 \cdot \mathbf{e}_1)\mathbf{e}_1|}$$

$$\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$$

The **Coordinate System Settings** node's settings window includes the following section:

SETTINGS

Here you define the direction of the second basis vector $\mathbf{v}_2$ by selecting one of the directions from the **Second basis vector** list: **x-axis**, **y-axis**, **z-axis** (in 3D), or **User defined**. If you select **User defined**, enter the components of $\mathbf{v}_2$ in the corresponding text fields under **Second basis vector**. You can select any other coordinate system for the Component from the **Coordinate system** list to use as the coordinate system for defining the second basis vector. The **Global coordinate system** is the default.

Using Extra Dimensions

To enable the **Extra Dimensions** option, go to The Preferences Dialog Box. Click **Model Builder**, select the **Enable technology preview functionality** check box, and then click **OK**.

Extra dimensions can be used to extend a standard geometry with additional dimensions. Using extra dimensions it is possible, in principal, to solve PDEs in any number of independent variables.

- The Component Node
- Adding Extra Dimensions to a Component
- Definitions

1 Enable the feature from the **Preferences>Model Builder** dialog box. Select the **Enable technology preview functionality** check box. Click **OK**.
2 To add an extra dimension, right-click the The Component Node. See Adding Extra Dimensions to a Component.
3 From the Definitions node, attach the extra dimensions to a selection in the base geometry. See Attached Dimensions.
4 As required, add and define Points to Attach and Integration Over the Extra Dimension nodes.
5 Adjust the settings for these topics: Defining Equations and Variables on Extra Dimensions, Selections in the Geometry, and Plotting Results in Extra Dimensions.

DEFINING EQUATIONS AND VARIABLES ON EXTRA DIMENSIONS

To define equations in the product geometry formed by an Attached Dimensions feature, add a Weak Contribution (PDEs) feature to any physics. In the **Selection** section, select an extra dimension attachment feature in the **Extra dimensions to attach** table, and make a selection of geometric entities in the base geometry and in each attached extra dimension.

By default, there are no dependent variables defined in the extra dimensions. To define dependent variables, add an Auxiliary Dependent Variable subfeature, and select the geometric entities in the base and extra dimension geometries where it should be defined.

Constraints in the extra dimensions can be defined by using Pointwise Constraint or Weak Constraint features with a selection in the product geometry.

SELECTIONS IN THE GEOMETRY

Whenever an extra dimension geometry has been attached using an Attached Dimensions feature, an **Extra dimensions to attach** list displays in the selection section for features that support selection in the product geometry. By default the extra dimension attachment is set to **None**.

If the **Extra dimensions to attach** setting is changed to one of the **Attached Dimension** features, additional inputs appear for each attached extra dimension geometry. Use these to choose the geometric entity level and the geometric entities to select in each extra dimension.

Features that currently support selection in the product geometry are **Variables**, Weak Contribution (PDEs), Auxiliary Dependent Variable, Pointwise Constraint, and Weak Constraint.

PLOTTING RESULTS IN EXTRA DIMENSIONS

A solution obtained by means of extra dimensions can be plotted in several ways:

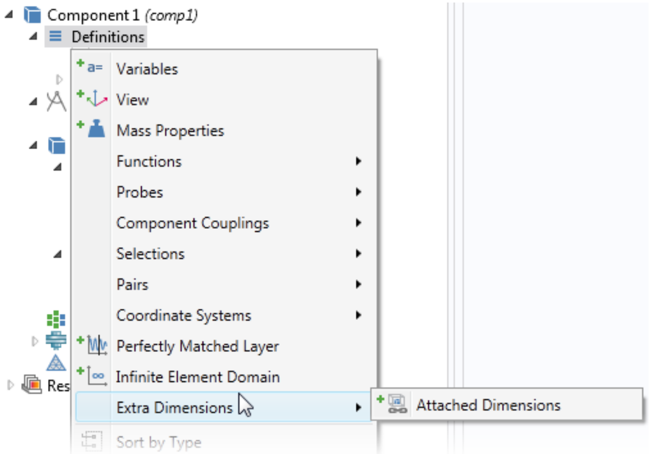
- A “horizontal” section through the product geometry can be plotted by using one of the `atxd` operators. For example, if a 2D extra dimension with tag `xdim1` is attached to a Component with tag `comp1`, the operator `comp1.xdim1.atxd2(x0,y0,expr)` evaluates `expr` at a point in the product geometry, defined by the coordinates (x0,y0) in the extra dimension geometry.
- Integrals over sections through the product geometry can be computed by using operators defined by Integration Over the Extra Dimension features. For example, if an integration operator called `xdintop1` has been defined, `comp1.xdim1.xdintop1(expr)` integrates `expr` over sections through the product geometry corresponding to the operator’s selection of geometric entities in the extra dimension geometry.
- It is also possible to make plots to plot along “vertical” sections through the product geometry. With Data Sets, select the extra dimension as Component. Then, for example if the base geometry is in 3D, evaluate `comp1.atxd3(x0,y0,z0,expr)` where (x0,y0,z0) define the coordinates of a point in the base geometry.

- Example: Solving Poisson’s Equation in a Cylinder by Means of Extra Dimensions
- Results Analysis and Plots
- Working with Geometric Entities
- Named Selections

Attached Dimensions

An **Attached Dimensions** ( ) node forms the Cartesian product of its selection in the base geometry, and the entire geometries of all selected extra dimensions. It is possible to add several Attached Dimensions features forming Cartesian products with different sets of extra dimensions. Each such Cartesian product is called a *product geometry*.

Before Extra Dimensions can be used in physics, it must be attached on a selection in the base geometry. Under the base **Component** node, right-click **Definitions** and from the **Extra Dimensions** submenu, select **Attached Dimensions**.



Example: Solving Poisson’s Equation in a Cylinder by Means of Extra Dimensions

GEOMETRIC ENTITY SELECTION

From the **Geometric entity level** list, choose **Entire geometry** (the default), **Domain**, **Boundary**, **Edge** (3D), or **Point** to determine where extra dimensions should be attached.

Click the **Active** button to toggle between turning ON and OFF selections. For Windows users, the buttons are ON □ and □ OFF . For Mac and Linux users the buttons are (⏻) for ON, and (⏻) for OFF.

ATTACHED DIMENSIONS

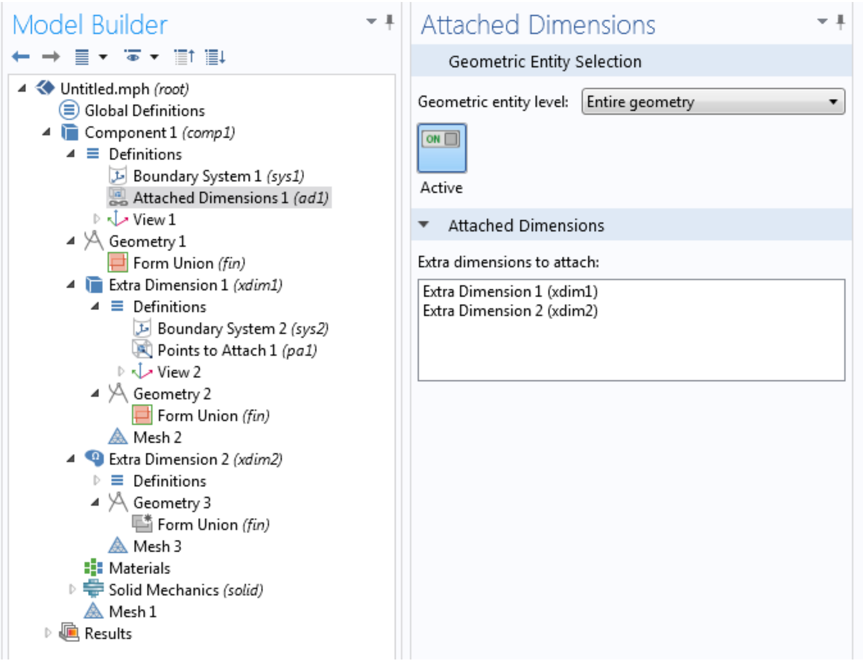
The **Extra dimensions to attach:** table lists all the Extra Dimension features included with the base Component node. See Figure 16-4.

A product geometry is formed as the Cartesian product of the selection of geometric entities in the base geometry and the entire geometry in all selected extra dimensions.

When the **Attached Dimensions** ( ) node is used in the **Extra dimensions to attach** by some selection in the model, it is not possible to make changes in the list of attached dimensions.

To make a change in the list of attached dimensions, change the **Extra dimensions to attach** list in all selections using the **Attached Dimensions** feature to some other **Attached Dimensions** feature or to **None**.

Deleting an **Attached Dimensions** node from the model resets **Extra dimensions to attach** to **None** for all selections using this **Attached Dimension** node.



Geometric entity level: Entire geometry

Extra dimensions to attach:

Extra Dimension 1 (xdim1)
Extra Dimension 2 (xdim2)

Figure 16-4: The Attached Dimension feature lists all the Extra Dimensions added to a Component. In this example, there are 3D and 2D Extra Dimensions under a 3D Component. Note that there is also a Definitions node under Extra Dimensions that contains a default Points to Attach feature.

Points to Attach

In each extra dimension, there is a **Points to Attach** ( ) node added by default to the **Definitions** node under Extra Dimension (See Figure 16-4). This can be used to select one or several points in the extra dimension geometry which are identified with the base geometry. In other words, the base geometry is identified with one or several sections through the product geometry.

In many cases, it is not necessary to select any points to attach.

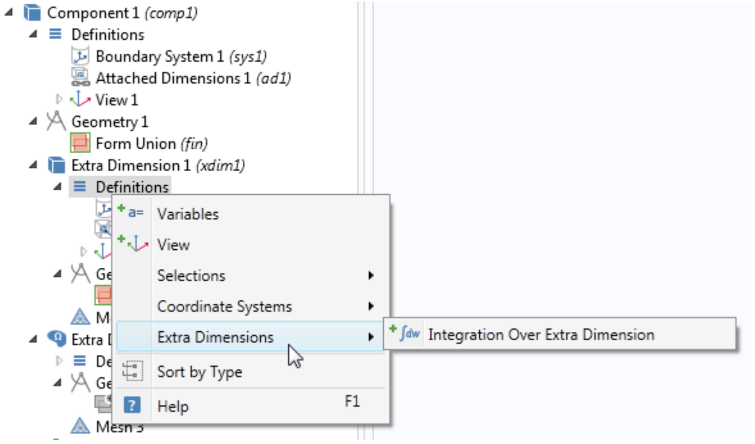
POINTS TO ATTACH

Select a set of points in the extra dimension. These points define sections through the product geometry that are identified with the base geometry.

Example: Solving Poisson's Equation in a Cylinder by Means of Extra Dimensions

Integration Over the Extra Dimension

In the **Integration Over Extra Dimension** ( ) node, make a selection of geometric entities to integrate over. To create an operator for integration over an extra dimension, under the Extra Dimension node, right click **Definitions** and from the **Extra Dimensions** submenu, choose **Integration Over Extra Dimension**.



OPERATOR NAME

Enter the **Operator name** of the integration operator.

GEOMETRIC ENTITY SELECTION

Select the geometric entities in the extra dimension geometry to integrate over.

ADVANCED

Select the **Integration order** and the **Frame** to integrate in. If the Extra Dimension is an axisymmetric geometry, select if the integral should be computed in the revolved geometry.

Example: Solving Poisson's Equation in a Cylinder by Means of Extra Dimensions

---

Example: Solving Poisson's Equation in a Cylinder by Means of Extra Dimensions

It is straightforward to solve Poisson's equation in a cylinder by setting up a 3D model in COMSOL. This example illustrates the use of extra dimensions by explaining how to solve the same problem by using a 2D base geometry with a 1D extra dimension.

- Creating a New Model
- Introduction to Solvers and Studies
- Creating a Geometry for Analysis

---

1 In the Model Wizard, create a 2D model and add a Weak Form PDE, and a **Stationary** study.

2 In the geometry, draw a **Circle**.

3 Add an extra dimension. Right-click **Component 1** and from the **Add Extra Dimension** menu, select **1D**.

4 In the extra dimension geometry, **Geometry 2** under **Extra Dimension 1**, draw an **Interval**.

5 Attach the extra dimension. Under **Component 1**, right-click **Definitions** and from the **Extra Dimensions** submenu, select **Attach Dimensions** ( ).

6 In the settings window for **Attached Dimensions 1**, click **Extra Dimension 1** in the **Extra dimensions to attach** table.

Attaching this extra dimension forms a Cylinder as a Cartesian product of the circle in the base geometry and the interval in the extra dimension geometry.

---

7 Add an equation on the product geometry. From the **Physics** toolbar, **Domains** menu, click **Weak Contribution**.

8 In the **Weak Contribution** settings window under **Domain Selection**, select **All domains**.

9 In the **Extra dimensions to attach** list under **Domain Selection**, select **Attached Dimensions 1**. In the **Extra Dimension 1** selection that appears, select **All domains**. This selects the entire product geometry for the weak contribution.

10 In the **Weak expression** text box enter the expression

```
-test(vx)*vx-test(vy)*vy-test(vx1)*vx1+1[m^-2]*test(v)
```

which is a weak-form expression for Poisson's equation.

11 To add a dependent variable on the product geometry, right- click **Weak Contribution 1** and select **Auxiliary Dependent Variable**.

12 In the **Domain selection** for **Auxiliary Dependent Variable 1**, select **Attached Dimensions 1** in the **Extra dimension attachment** list. In the **Extra Dimension 1** selection that appears, select **All domains**. This selects the entire product geometry for the auxiliary dependent variable. In the **Field variable name** text field enter the variable name `v`.

13 To add boundary conditions, right-click **Weak Form PDE** and select **More>Pointwise Constraint** (in the boundary section). In **Boundary selection** for **Pointwise Constraint 1**, select **All boundaries**, and in the **Extra dimension attachment** list select **Attached Dimensions 1**. In the **Extra Dimension 1** selection that appears, select **All domains**. In

the **Constraint expression** check box enter the expression `-v`. This applies a constraint $v = 0$ to `v` on the curved surfaces of the cylinder.

14 To add constraints also on the top and bottom of the cylinder, right click **Weak Form PDE** and select **More> Pointwise Constraint** (in the domain section). In **Domain selection** for **Pointwise Constraint 2**, select **All domains**, and in the **Extra dimension attachment** list select **Attached Dimensions 1**. In the **Extra Dimension 1** selection that appears, set **Geometric entity level** to **Boundary** and **Selection** to **All boundaries**. In the **Constraint expression** text field enter the expression `-v` to constrain v to 0.

15 Before the model can be solved it is necessary to add suitable boundary conditions for the default dependent variable `u` (defined only in the base geometry). Right click **Weak Form PDE**, select **Dirichlet Boundary Condition**, and in **Dirichlet Boundary Condition 1** select **All boundaries**. By default this boundary condition sets u to 0.

16 Right-click **Study 1** and select **Compute**.

By default a plot of the dependent variable `u`, defined on the base geometry, is shown.

1 To plot `v`, go to **2D Plot Group 1>Surface 1** and enter the expression `comp1.xdim1.atxd1(0.4,v)`. This makes a plot of `v` on a slice through the cylinder at height 0.4. The `atxd`*`i`* operator evaluates an expression at a location in the extra dimension.

2 To make a plot of `v` along a vertical line, right-click the data set **Solution 1** and select **Duplicate**.

3 In the new data set **Solution 2**, select **Extra Dimension 1** from the **Component** list.

4 Right-click **Results** and select **1D Plot Group**. In **1D Plot Group 2** select **Solution 2** from the **Data set** list.

5 Right-click **1D Plot Group 2** and select **Line Graph**. In **Line Graph 1**, set the selection to **All domains**, and under **y-Axis Data** enter the expression `comp1.atxd2(0.2,0.5,v)`, and click **Plot**. This plots `v` along a vertical line above the point with coordinates (0.2,0.5) in the base geometry.

17

Sensitivity Analysis

This chapter describes how to perform sensitivity analysis using the Sensitivity interface, which is found under the **Mathematics>Optimization and Sensitivity** (🎯) branch when adding an interface.

Theory for the Sensitivity Interface

About Sensitivity Analysis

The Sensitivity interface is special in the sense that it does not contain any physics of its own. Instead, it is a tool for evaluating the sensitivity of a model with respect to almost any variable. The Sensitivity interface is used together with a Sensitivity study step, which in turn controls the Sensitivity solver extension. Simple cases can be handled directly in the Sensitivity study step, while more advanced cases must be set up in a Sensitivity interface prior to solving.

Simulation is a powerful tool in science and engineering for predicting the behavior of physical systems, particularly those that are governed by partial differential equations. However, a single simulation is often not enough to provide sufficient understanding of a system. Hence, a more exploratory process might be needed, such as *sensitivity analysis*, where one is interested in the sensitivity of a specific quantity with respect to variations in certain parameters included in the model. Such an analysis can, for example, be used for estimating modeling errors caused by uncertainties in material properties or for predicting the effect of a geometrical change.

Many times it is possible to reformulate problems of the above type as the problem of calculating derivatives, so differentiation plays a central role in solving such problems. The Sensitivity study step and corresponding interface can calculate derivatives of a scalar *objective function* with respect to a specified set of *control variables*. The objective function is in general a function of the solution to a multiphysics problem, which is in turn parameterized by the control variables.

Sensitivity Problem Formulation

Because the Sensitivity interface does not contain any physics, it is not intended for use on its own. When the interface is added to a multiphysics model, no new equations are introduced, and the set of solution variables remains the same. Instead, an objective function and a set of control variables can be specified. The Sensitivity interface can perform these distinct tasks:

- Select control variables and set their values
- Define scalar objective functions

The control variables are independent variables whose values are not affected by the solution process, but they are also degrees of freedom (DOFs) stored in the solution vector. When defining a control variable, its *initial value* must be supplied. The initial value is used to initialize the control variable DOFs, which remain fixed during the solution step.

The companion Sensitivity study step is responsible for:

- Choosing which objective functions and control variables to solve for
- Selecting a sensitivity evaluation method
- Selecting which study step to compute sensitivities for
- Setting up the Sensitivity solver extension

Evaluating the sensitivity of a scalar-valued objective function $\boldsymbol{Q}(\xi)$ with respect to the control variables, ξ, at a specific point, ξ_0, can be rephrased as the problem of calculating the derivative $\partial \boldsymbol{Q}/\partial\xi$ at $\xi = \xi_0$. In the context of a multiphysics model, $\boldsymbol{Q}$ is usually not an explicit expression in the control variables ξ alone. Rather, $\boldsymbol{Q}(u(\xi), \xi)$ is also a function of the solution variables u, which are in turn implicitly functions of ξ.

The multiphysics problem is a PDE, which after discretization is represented as a system of equations $\boldsymbol{L}(u(\xi), \xi) = 0$. If the PDE has a unique solution $u = \boldsymbol{L}^{-1}(\xi)$, the sensitivity problem can be informally rewritten using the chain rule as that of finding

$$\frac{d}{d\xi}\boldsymbol{Q}(u(\xi), \xi) = \frac{\partial \boldsymbol{Q}}{\partial \xi} + \frac{\partial \boldsymbol{Q}}{\partial u}\cdot\frac{\partial u}{\partial \boldsymbol{L}}\cdot\frac{\partial \boldsymbol{L}}{\partial \xi}$$

The first term, which is an explicit partial derivative of the objective function with respect to the control variables is easy to compute using symbolic differentiation. The second term is more difficult. Assuming that the PDE solution has N degrees of freedom and that there are n control variables ξ_i, $\partial\boldsymbol{Q}/\partial u$ is an N-by-1 matrix, $\partial u/\partial \boldsymbol{L}$ is an N-by-N matrix (since $\boldsymbol{L}^{-1}$ is unique) and $\partial \boldsymbol{L}/\partial\xi$ is an N-by-n matrix.

The first and last factors, $\partial\boldsymbol{Q}/\partial u$ and $\partial \boldsymbol{L}/\partial\xi$ can be computed directly using symbolic differentiation. The key to evaluating the complete expression lies in noting that the middle factor can be computed as $\partial u/\partial \boldsymbol{L} = (\partial \boldsymbol{L}/\partial u)^{-1}$ and that $\partial \boldsymbol{L}/\partial u$ is the PDE Jacobian at the solution point:

$$\frac{d}{d\xi}\boldsymbol{Q}(u(\xi), \xi) = \frac{\partial \boldsymbol{Q}}{\partial \xi} + \frac{\partial \boldsymbol{Q}}{\partial u}\cdot\left(\frac{\partial \boldsymbol{L}}{\partial u}\right)^{-1}\cdot\frac{\partial \boldsymbol{L}}{\partial \xi} \tag{17-1}$$

Actually evaluating the inverse of the N-by-N Jacobian matrix is too expensive. In order to avoid that step, an auxiliary linear problem can be introduced. This can be done in two different ways, each requiring at least one additional linear solution step:

FORWARD SENSITIVITY METHOD

To use the *forward sensitivity* method, introduce the N-by-n matrix of solution sensitivities

$$\frac{\partial u}{\partial \xi} = \left(\frac{\partial \boldsymbol{L}}{\partial u}\right)^{-1}\cdot\frac{\partial \boldsymbol{L}}{\partial \xi}$$

These can be evaluated by solving n linear systems of equations

$$\frac{\partial \boldsymbol{L}}{\partial u}\cdot\frac{\partial u}{\partial \xi_i} = \frac{\partial \boldsymbol{L}}{\partial \xi_i}$$

using the same Jacobian $\partial \boldsymbol{L}/\partial u$, evaluated at $u(\xi_0)$. Inserting the result into Equation 17-1, the desired sensitivities can be easily computed as

$$\frac{d}{d\xi}\boldsymbol{Q}(u(\xi), \xi) = \frac{\partial \boldsymbol{Q}}{\partial \xi} + \frac{\partial \boldsymbol{Q}}{\partial u}\cdot\frac{\partial u}{\partial \xi}$$

ADJOINT SENSITIVITY METHOD

To use the *adjoint sensitivity* method, introduce instead the N-by-1 adjoint solution u^*, which is defined as

$$u^* = \frac{\partial \boldsymbol{Q}}{\partial u}\cdot\left(\frac{\partial \boldsymbol{L}}{\partial u}\right)^{-1}$$

Multiplying this relation from the right with the PDE Jacobian $\partial \boldsymbol{L}/\partial u$ and transposing leads to a single linear system of equations

$$\frac{\partial L}{\partial u}^T \cdot u^* = \frac{\partial Q}{\partial u}$$

using the transpose of the original PDE Jacobian.

Specification of the Objective Function

The objective function can in general be a sum of a number of terms:

$$Q(u, \xi) = Q_{\text{global}}(u, \xi) + Q_{\text{probe}}(u, \xi) + \sum_{k=0}^{n} Q_{\text{int},\, k}(u, \xi)$$

where n is the space dimension of the multiphysics model and the different contributions in the sum above are defined as follows:

- Q_{global} is the *global contribution* to the objective function Q. It is given as one or more general global expressions.
- Q_{probe} is a *probe contribution* to the objective function Q. It is a *probe objective* so its definition is restricted to a point on a given geometrical entity. The *probe point* used for the point evaluation is a point given by the user and has to be contained in the domain.
- $Q_{\text{int},k}$ is an *integral contribution* to the objective function Q. It is an *integral objective* so its definition is restricted to a specific set of geometrical entities of the same dimension. For integral contributions on points, the integration reduces to a summation.

Several global, probe, and integral contributions can be defined. In such cases, the total global, probe, and integral contribution is given as the sum of the aforementioned global, probe, and integral contributions that are actively selected in the solver settings for the optimization.

Choosing a Sensitivity Method

To evaluate sensitivities as part of a multiphysics problem solution, an auxiliary linear problem must be solved, in addition to the original equation, using one of these methods:

- Select one of the Forward Sensitivity methods to evaluate the derivatives of all solution variables and an optional objective function.
- Select the Adjoint Sensitivity method to look only at derivatives of a scalar objective function.

FORWARD SENSITIVITY

Use the forward (or forward numeric) sensitivity method to solve for the derivatives of all dependent variables, plus an optional scalar objective function, with respect to a small number of control variables. The forward method requires one extra linear system solution for each control variable.

The linear system that must be solved is the same as the last linearization needed for solving the forward model. Thus, when using a direct solver (for example, PARDISO) the extra work amounts only to one back-substitution per control variable DOF. The forward numeric method in addition requires two additional residual evaluations. The iterative linear and segregated solvers can reuse preconditioners and other data but must otherwise perform a complete solution each time.

ADJOINT SENSITIVITY

The adjoint method solves for the derivatives of a single scalar objective function with respect to any number of control variables, requiring only one single additional linear system solution. In addition to the objective function gradient, the discrete adjoint solution is computed. This quantity represents the sensitivity of the objective function with respect to an additional generalized force applied as a nodal force to the corresponding solution component.

The auxiliary linear system is in this case the transpose of the last linearization needed for solving the forward model. The MUMPS and PARDISO linear solvers can solve the transposed problem at the cost of a back-substitution, while the SPOOLES linear solver needs to do a new factorization if the problem is not symmetric or Hermitian. The iterative solvers can reuse most preconditioning information as can the segregated solver, which, however, loops over the segregated steps in reversed order.

Sensitivity analysis can be used together with all stationary and parametric standard solvers and with the BDF solver for transient studies. The available solvers are described in the section Studies and Solvers.

Postprocessing Sensitivities

When a multiphysics problem is solved using sensitivity analysis, the generated solution will contain stored sensitivity data. You can access this data in postprocessing using the `fsens` and `sens` operators:

- `fsens(<control_variable>)` evaluates the sensitivity (derivative) of the objective function with respect to the specified control variable. This result is available for all sensitivity methods. The result of `fsens` is a global quantity which can be evaluated in a Global Evaluation feature or on any geometric entity.
- `sens(<dependent_variable>,<control_variable>)` evaluates the sensitivity (derivative) of the specified dependent variable with respect to the specified control variable. This is only possible when forward sensitivity has been used, which computes and stores derivatives of the entire solution vector with respect to each control variable degree of freedom. The result of `sens` has the same geometric scope as the dependent variable argument; it can be plotted or evaluated wherever the dependent variable itself is available.

Issues to Consider Regarding the Control Variables

THE EFFECT OF DISCRETIZATION

The sensitivity analysis is always performed on the discretized system of equations. As already mentioned, the control variables can be a scalar, vector, or an element in some infinite-dimensional function space. In the latter case it is represented on the finite element mesh, just like the solution variables, or global scalar quantities. When using a control variable field represented on the finite element mesh, the sensitivities are therefore associated with individual control variable degrees of freedom rather than with the field value at each point. This makes it difficult to interpret the result. For example, if a domain control variable is set up using a first-order Lagrange shape function representation to control the material density in a model, the solution contains the sensitivity of the objective function with respect to the discrete density value at each node point in the mesh. Because each node influences the density in a small surrounding region, the size of which varies from node to node, the individual sensitivities are not directly comparable to each other.

Displaying such domain control variables results in a plot that is not smooth due to the varying element size. It must therefore not be used to draw any conclusions about the physics and the effect of changing the physical field represented by the control variable. Some insight can, however, be gained by looking at the sensitivities divided by the mesh volume scale factor `dvol`. This makes the sensitivities in the plot comparable between different parts of the surface but still not mathematically well defined. In particular, using discontinuous constant shape functions together with the division by `dvol` results in a plot that is proportional to the true pointwise sensitivity.

> If the plan is to use the sensitivities in an automatic optimization procedure, as is done through the Optimization interface available with the Optimization Module, the discrete nature of the sensitivities causes no additional complication. The optimization solver searches for optimum values of the discrete control variables using the discrete gradient provided by the sensitivity analysis.

GEOMETRICAL SENSITIVITY

You can use the control variables directly to parameterize any aspect of the physics that is controlled by an expression. This applies to material properties, boundary conditions, loads, and sources. However, the shape, size, and position of parts of the geometry cannot be changed as easily at solution time and require special attention.

Control variables cannot be used directly in the geometry description. Instead, the model must be set up using a Deformed Geometry or Moving Mesh interface to control the shape of the geometry. Then use control variables to control the mesh movement, effectively parameterizing the geometry. See Deformed Geometry and Moving Mesh for details about these interfaces and ALE in general.

Issues to Consider Regarding the Objective Function

THE PRINCIPLE OF VIRTUAL WORK

Potential energy has a special status among scalar objective functions, because its derivatives with respect to scalar control variables can in many cases be interpreted as (true or generalized) forces.

COMPLEX-VALUED OBJECTIVE FUNCTIONS

Sensitivity analysis can be applied only when the objective function is a real differentiable or complex analytic function of the control variables. This is usually not a very severe constraint, even for frequency-domain models where the PDE solution variables are complex valued. The reason is that physical quantities of interest to the analyst are always real-valued, and if complex-valued control variables are required, it is possible to treat the real and imaginary parts separately.

Many common quantities of interest are time averages which can be written in the form $Q = \mathrm{real}(a \cdot \mathrm{conj}(b))$, where a and b are complex-valued linear functions of the solution variables and therefore implicit functions of the control variables. The problem with this expression is that while Q is indeed a real-valued differentiable function of the control variables, it is not an analytical function of a and b. This complicates matters slightly because the sensitivity solver relies on partial differentiation and the chain rule.

While the partial derivatives of Q with respect to a and b are, strictly speaking, undefined, it can be proven that if they are chosen such that

$$Q(a + \delta a, b + \delta b) \approx Q(a, b) + \mathrm{real}\left(\frac{\partial Q}{\partial a}\delta a + \frac{\partial Q}{\partial b}\delta b\right) \tag{17-2}$$

for any small complex increments δa and δb, the final sensitivities are evaluated correctly. The special function `realdot(a,b)` is identical to `real(a*conj(b))` when evaluated but implements partial derivatives according to Equation 17-2. For that reason, use it in the definition of any time-average quantity set as objective function in a sensitivity analysis.

The Sensitivity Interface

The **Sensitivity (sens)** interface (), found under the **Mathematics>Optimization and Sensitivity** () branch when adding a physics interface, provides tools for adding advanced sensitivity evaluation to a stationary model. Basic problems only in terms of global scalar objective functions and model parameters can be set up directly in a Sensitivity study step, and therefore do not require the use of a Sensitivity interface.

For a more extensive introduction to the mathematics implemented by this interface, see the Theory for the Sensitivity Interface.

The objective functions are defined in terms of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model), which can be fields dependent on position in space or scalar quantities defined globally. This flexibility is reflected in the interface by grouping these settings according to the dimension of the domain to which they apply. In such a group of settings, the following settings can be specified, to each of which corresponds a separate feature and its settings window:

- Integral Objective
- Probe Objective
- Control Variable Field

Note that adding a Sensitivity study step to a study makes it possible to perform a sensitivity analysis directly at the study level. See Sensitivity .

The main **Sensitivity** node's settings window contains the following section:

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `sens`.

- Global Objective
- Global Control Variable

Sensitivity Analysis of a Communication Mast Detail: model library path **COMSOL_Multiphysics/Structural_Mechanics/mast_diagonal_mounting_sensitivity**

Integral Objective

An **Integral Objective** is defined as the integral of a closed form expression of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model) that are either global or available in the domain in question. Hence, its definition is restricted to a set of geometric entities of the same dimension. For integral objectives on points, the integration reduces to a summation.

DOMAIN, EDGE, BOUNDARY, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) used in the integration for the integral objective.

OBJECTIVE

Enter an **Objective expression** that is integrated over the geometric entity level in the integral objective.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

QUADRATURE SETTINGS

Specify the settings for the **Quadrature** used to numerically evaluate the integral in the integral objective: the integration order (default: 4) in the **Integration order** field and the frame to integrate on (default: the spatial frame), which is selected from the **Integrate on frame** list.

Probe Objective

A **Probe Objective** is defined as a point evaluation of a closed form expression of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model) that are either global or available in the domain in question. The point used for the point evaluation has to be contained in the domain.

DOMAIN SELECTION

From the **Selection** list, choose the domain containing the point used for the point evaluation.

OBJECTIVE

Enter an **Objective expression** that is evaluated at the point in the domain.

PROBE COORDINATES

Specify the **Probe coordinates** for the point in the domain where the expression for the objective is evaluated. After specifying the probe coordinates, select an option from the **Evaluate in frame**—**Spatial** (the default), **Material**, or **Mesh**.

Control Variable Field

Specify the **Control Variable Field** specific to the geometric entity level (domain, edge, boundary, or point) in question.

DOMAIN, EDGE, BOUNDARY, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

- Continuity on Interior Boundaries
- Identity and Contact Pairs

CONTROL VARIABLE

Enter a **Control variable name** and **Initial value**.

DISCRETIZATION

This section contains settings for the element used to discretize the control variable. Select a **Shape function type**: **Lagrange** (the default) or **Discontinuous Lagrange**. Also select an **Element order**: **Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**.

Global Objective

Specify the **Global Objective** contribution to the function by entering an objective expression.

OBJECTIVE

Enter an **Objective expression** that defines the contribution to the objective function. It can be an expression of those components of the control and solution variable (the solution variable is given as the solution to the differential equations defined by the multiphysics model) that are globally available.

Global Control Variable

Use the **Global Control Variable** node to specify any globally available control variables.

CONTROL VARIABLES

In the table, enter **Variable** names and **Initial values** of the control variables that are globally available. To add a control variable to the table, click the **Add** button (✚). To remove a control variable and its values from the table, click the **Delete** button (✖).

18

Deformed Geometry and Moving Mesh

This chapter explains how to use the interfaces that control mesh deformation, which you can find under the **Mathematics>Deformed Mesh** branch (▦) when adding physics interfaces to a model. It also contains fundamentals about deformed meshes and information about the Eulerian and Lagrangian formulations of the physics, the frame types that support these formulations, and the arbitrary Lagrangian-Eulerian (ALE) method.

Deformed Mesh Fundamentals

About Deformed Meshes

A deformed mesh can be useful if the boundaries of your computational domain are moving in time or deform as a function of some parameter. The deformation can also be physics induced—for example, depend on computed velocities or solid deformation. The point is that a new mesh need not be generated for each configuration of the boundaries—instead, the software perturbs the mesh nodes so they conform with the moved boundaries.

In COMSOL Multiphysics, control the movement of the interior nodes in these ways:

- By propagating the moving boundary displacement throughout the domain to obtain a smooth mesh deformation everywhere. This is done by solving PDEs for the mesh displacements (a Laplace, Winslow, or hyperelastic smoothing PDE, or one borrowed from continuum mechanics) with boundary conditions given by the movement of the boundaries.
- By specifying an explicit formula for the mesh deformation. The formula can make use of other dependent variables, such as the displacement components of structural mechanics.
- By leaving the control of the mesh displacement to a Solid Mechanics interface, which has built-in deformed mesh functionality, or to a multiphysics interface of which Solid Mechanics is part.

Deformed Geometry vs. Moving Mesh

There are two interfaces implementing different types of deformed meshes, both selected from under the **Mathematics>Deformed Mesh** branch (▦): the **Deformed Geometry** (▣▣) and the **Moving Mesh** (▦) interfaces.

In the Deformed Geometry interface, the material does not follow the change in shape. Deformation of the geometry boundaries therefore corresponds to addition or removal of material. In the Moving Mesh interface, solid materials follow the mesh deformation and deform in the same way as the mesh. Fluids and gases, on the other hand, are added or removed so as to always fill the current shape of the domain—any effects of compression or expansion must be introduced explicitly into the equations.

- Use a Deformed Geometry interface to study the behavior of different shapes of an original object. In a Deformed Geometry interface the material never follows a perturbation of the shape. The total mass of the first shape is not the same as the mass for the second, perturbed, geometry. Any deformation can be regarded as removal or addition of material.
- Use a Moving Mesh interface to study how a solid object deforms as the results of physical load, and how fluids in adjacent domains react to displacement of the domain boundaries—for example, how a tank impeller moves a fluid, or how a MEMS switch moves under the influence of an electric field. Using the Moving Mesh interface, a solid material follows the mesh deformation. A movement of a boundary can therefore be regarded as bending or punching the original object. Undeformed and deformed solid objects have the same mass, but the total amount of fluid in a domain whose boundaries deform can change.

Arbitrary Lagrangian-Eulerian Formulation (ALE)

The partial differential equations of physics are usually formulated either in a *spatial* coordinate system, with coordinate axes fixed in space, or in a *material* coordinate system, fixed to the material in its reference configuration and following the material as it deforms. The former is often referred to as an *Eulerian* formulation, while the latter is a *Lagrangian* formulation.

Structural mechanics and other fields of physics dealing with a possibly anisotropic, solid, material are most conveniently simulated using material coordinates. The Lagrangian formulation makes the anisotropic material properties independent of the current spatial orientation of the material.

If, on the other hand, the focus is on simulating the physical state at fixed points in space, an Eulerian formulation is usually more convenient. In particular, when liquids and gases are involved, it is often unreasonable to follow the state of individual material particles. Rather, the quantities of interest are pressure, temperature, concentration, and so forth, at fixed positions in space.

An inherent problem with the pure Eulerian formulation is that it cannot handle moving domain boundaries, since physical quantities are referred to fixed points in space, while the set of spatial points currently inside the domain boundaries changes with time. Therefore, to allow moving boundaries, the Eulerian equations must be rewritten so as to describe all physical quantities as functions of some coordinate system in which the domain boundaries are fixed. The finite element mesh offers one such system: the *mesh* coordinates.

In the mesh coordinate system, the domain is fixed, and there is a one-to-one map from the mesh coordinates to the current spatial configuration of the domain. Otherwise, the mesh coordinate system can be defined freely and separately from both the spatial and material systems. The natural choice is to let the mesh coordinate system, at least initially, coincide with the *geometry* coordinates. This follows immediately from the way meshes are created, and means that points in the domain are identified by their position in the original geometry.

As the domain and mesh deform, the map from mesh coordinates to spatial coordinates can become increasingly ill-conditioned. Before the degradation of the mesh mapping goes too far you can, using a remeshing operation, stop the simulation, create a new mesh in the current configuration of the domain, and map all quantities to the new mesh. When you restart simulation, points in the domain are internally identified by their new mesh coordinates, which coincide with the spatial coordinates at the state where the simulation was stopped. Therefore, the geometry and mesh coordinates of a given point differ after remeshing the deformed geometry.

Rewriting physics equations in this way, on a freely moving mesh, results in an *arbitrary Lagrangian-Eulerian* (ALE) method. In the special case when the map from mesh coordinates to spatial coordinates follows the material deformation, a Lagrangian method is recovered. Similarly, when the map is an identity map, the ALE method becomes entirely Eulerian.

The ALE method is therefore an intermediate between the Lagrangian and Eulerian methods, and it combines the best features of both—it allows moving boundaries without the need for the mesh movement to follow the material.

About Frames

COMSOL Multiphysics refers to the spatial, material/reference, geometry, and mesh coordinate systems described above as *spatial frame*, *material frame* (reference frame), *geometry frame*, and *mesh frame*, respectively. Physics can be formulated on the spatial frame or on the material frame, depending on whether it is more convenient to interpret the equations as Eulerian or Lagrangian, respectively. It is not possible to use the geometry and mesh frames and their associated coordinates to formulate physics because they are neither connected to the material nor to the true Euclidean space.

Conceptually, all four frames always exist, but all or some of them can point to the same actual coordinate system. It is the actual coordinate system that decides the names of the independent variables (the coordinate names like `x`, `y`, `z` or `r`, `phi`, `z`). Before adding a Moving Mesh or Deformed Geometry interface to a Component, all four frames coincide and use the spatial coordinate names. Also all physics interfaces based on solid mechanics include moving mesh functionality and by default behave much in the same way as a Moving Mesh interface.

When a Moving Mesh or Solid Mechanics interface is added, the spatial frame is separated from the material frame, which is given a new set of independent variable names (by default capital `X`, `Y`, `Z` or `R`, `PHI`, `Z`). From this point, Eulerian and Lagrangian formulations behave differently because they, among other things, define derivatives with respect to different sets of independent variables.

The geometry frame and the material frame use the same coordinate system until a Deformed Geometry interface is added. At that point, a new geometry coordinate system is created and given a new set of independent variable names (by default `Xg`, `Yg`, `Zg` or `Rg`, `PHIg`, `Zg`). The new geometry frame refers to the geometry as it is represented by the Geometry Sequence. By inserting a nontrivial transformation from geometry coordinates to material coordinates, the shape of the geometry can be effectively changed without having to create a new mesh. This can be useful as a means of parameterizing the geometry, for example, before performing optimization or sensitivity analysis.

Using Deformed Geometry affects both Eulerian and Lagrangian physics in the same way. The reason is that the Deformed Geometry interface controls the material frame in relation to the geometry frame. Unless there is also a Moving Mesh or Solid Mechanics interface present, the material frame and the spatial frame still refer to the same coordinate system. The three frames refer to three different sets of coordinates only when there is both a Deformed Geometry and some Moving Mesh interface active in the Component.

The geometry frame and the mesh frame coincide until a manual or automatic remeshing operation is performed. At that point, a new mesh is created in the original geometry together with a new set of coordinates (independent variable typically `Xm`, `Ym`, `Zm` or `Rm`, `PHIm`, `Zm`). The original geometry coordinates are mapped and stored together with the new mesh such that any Deformed Geometry interface can still define the material frame relative to the original geometry frame.

To avoid confusion, note that:

- The *spatial frame* is the usual, fixed, global, Euclidean coordinate system with the *spatial coordinates* (x, y). In the ALE context, the spatial coordinate system as such is fixed while the spatial coordinates (x, y) of each material point and mesh node can be functions of time. Therefore, it is correct to refer to the model as having a *moving mesh*.
- The *material frame* is a coordinate system which identifies material points by their spatial coordinates (X, Y) in some—actual or imagined—reference configuration. Think of the material coordinate system as having been printed on the material in the reference configuration such that it follows it during deformation. It is therefore in general curvilinear and cannot be used directly to measure true distances and angles. See also Figure 18-1 and Figure 18-2.
- The *geometry frame* is a coordinate system which identifies points by their spatial coordinates (X_g, Y_g) in the original geometry. It is often natural to use the original geometry also as reference state to define material coordinates. Therefore, the geometry frame and material frame usually coincide. The only exception is when a Deformed Geometry interface is used to deform or parameterize the original geometry.
- The *mesh frame* is a coordinate system used internally by the finite element method. It identifies mesh points by their spatial coordinates (X_m, Y_m) at the time the mesh was created. The original mesh is always created based on the original geometry. Therefore, the mesh frame coincides with the geometry frame until a new mesh is created in the—then current—deformed configuration.

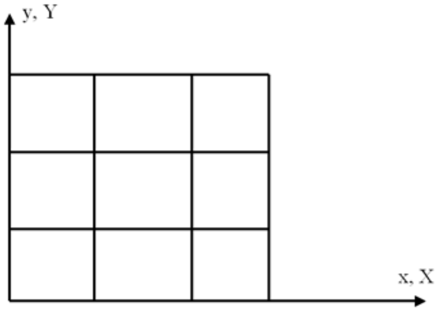


Figure 18-1: An undeformed mesh. In the reference configuration, which can be the actual configuration at a reference time or a hypothetical state, the spatial frame (x, y) and the material frame (X, Y) coincide.

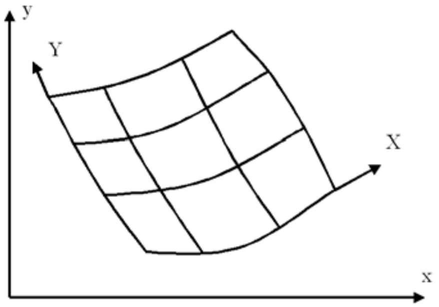


Figure 18-2: After deformation of the material, the spatial frame (x, y) remains the same, while the material coordinate system (X, Y) has been deformed, following the material. Meanwhile, the material coordinates of each material point remain the same but its spatial coordinates have changed.

Mathematical Description of the Mesh Movement

Though moving meshes are also possible in 3D, consider a 2D geometry for simplicity, where the spatial and material frame coordinates are called (x, y) and (X, Y), respectively. Let (X_0, Y_0) be the spatial coordinates of a mesh node in the initial material configuration. The spatial coordinates (x_0, y_0) of the same mesh node at some other time, t, are then given by the functions

$$x_0 = x(X_0, Y_0, t), \qquad y_0 = y(X_0, Y_0, t) \tag{18-1}$$

These functions can be explicit transformations (expressions) or the solution to a mesh smoothing equation. The mesh node's material coordinates (X_0, Y_0) can in turn be seen as functions of an underlying system of geometry coordinates $(X_\mathrm{g}, Y_\mathrm{g})$ and a parameter, p, such that

$$X_0 = X(X_g, Y_g, p), \qquad Y_0 = Y(X_g, Y_g, p) \tag{18-2}$$

with similar options for the transformations. The transformations can also be chained such that (x_0, y_0) are seen as functions of $(X_\mathrm{g}, Y_\mathrm{g})$, t, and p.

Introducing a vector notation for the coordinates:

- Spatial coordinates $\mathbf{x} = [x, y, z]$
- Material coordinates $\mathbf{X} = [X, Y, Z]$
- Geometry coordinates $\mathbf{X}_g = [X_g, Y_g, Z_g]$
- Mesh coordinates $\mathbf{X}_m = [X_m, Y_m, Z_m]$

the general relation between the frames can be written

$$\mathbf{x} = \mathbf{f}(\mathbf{X}, t) = \mathbf{f}(\mathbf{g}(\mathbf{X}_g, p), t)$$
$$\mathbf{X} = \mathbf{g}(\mathbf{X}_g, p)$$
$$\mathbf{X}_m = \mathbf{h}(\mathbf{X}_g, i)$$

where $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ are vector-valued functions, t is time, p is some set of parameters controlling a Deformed Geometry interface, and i is number of times the geometry has been remeshed. From the physics point of view, the domain is fixed in the geometry frame coordinates $\mathbf{X}_g$, which are therefore seen as constant in the above formulas.

From the finite elements' point of view, it is instead the mesh frame coordinates $\mathbf{X}_m$ that are constant and $\mathbf{X}_g = \mathbf{h}^{-1}(\mathbf{X}_m, i)$. Therefore when assembling the finite-element matrices, the relation actually used is

$$\mathbf{x} = \mathbf{f}(\mathbf{g}(\mathbf{h}^{-1}(\mathbf{X}_m, i), p), t)$$

where **f** is a unit map if the spatial and material frames coincide, **g** is a unit map if the material and geometry frames coincide, and the inverse mapping $\mathbf{h}^{-1}(\mathbf{X}_m, i)$ is initially a unit map and then updated by interpolation after each remeshing operation.

In addition to the different sets of coordinate variables, some other geometric variables that the software defines are available for both the spatial and the material frames (see Geometric Variables and Mesh Variables).

Derivatives of Dependent Variables

When solving for some physical quantity, u, COMSOL Multiphysics always stores the solution for a fixed set of mesh nodes. That is, the dependent variable u is treated internally as a function of the mesh coordinates, $u(X_m, Y_m, t)$. The essence of the ALE system is that it allows treating the physical quantities as functions of the material or spatial coordinates, $u(X, Y, t)$ or $u(x, y, t)$, instead. This transformation is possible only if the mappings given by Equation 18-1 and Equation 18-2 are invertible.

SPATIAL DERIVATIVES

With respect to spatial differentiation, each dependent variable is treated as a function of one or more of the frames present in the model. Most physics interfaces are based on a formulation which is either Eulerian or a Lagrangian. They therefore lock their dependent variables to the spatial or the material frame, respectively. A few physics can formulate their equations in either material or spatial frame, as set by the **Frame** setting found under **Discretization** in the physics interface node's settings.

For a dependent variable `u`, there are typically two possibilities:

- The variable is defined on the spatial frame and its derivatives with respect to the spatial coordinates are denoted `ux` and `uy` in the software.
- The variable is defined on the material frame and its derivatives with respect to the material coordinates are denoted `uX` and `uY` in the software.

In a few cases both sets of derivatives exist, but normally it is only possible to use one of these types of derivatives of each dependent variable.

TIME DERIVATIVES

When using ALE, the software defines two kinds of time derivatives:

- The common *frame time derivative*, valid for a fixed point in the frame on which the variable is defined. This derivative is always denoted `ut` in the software. For example, for `u` defined on the spatial frame:

$$u_t(x_0, y_0) = \left.\frac{\partial u}{\partial t}\right|_{x_0, y_0}$$

- The *mesh time derivative*, which is taken for a fixed point in the mesh:

$$u_{\mathrm{TIME}}(X_m, Y_m) = \left.\frac{\partial u}{\partial t}\right|_{X_m, Y_m}$$

  This derivative is denoted `uTIME` in the software. Since internally, everything is formulated on the mesh frame, the mesh time derivative is the one computed by the solvers and stored in the solution vector.

The two derivatives are related by the chain rule:

$$u_t = u_{\mathrm{TIME}} - u_x x_{\mathrm{TIME}} - u_y y_{\mathrm{TIME}}$$

where $(x_{\mathrm{TIME}}, y_{\mathrm{TIME}})$ is the mesh velocity. The mesh time derivative is often less important from the user point of view because its value depends on the mesh movement, which in itself has no physical significance. However, for the special case when the mesh follows the material's motion, the mesh time derivative is physically significant and is also called the *material time derivative*.

Smoothing Methods

In the domains with free displacement, the Moving Mesh interface solves an equation for the mesh displacement. This equation smoothly deforms the mesh given the constraints placed on the boundaries. Choose between *Laplace smoothing*, *Winslow smoothing*, *hyperelastic smoothing*, and *Yeoh smoothing*.

To specify the smoothing methods, use the **Mesh smoothing type** list in the **Free Deformation Settings** section of the **Moving Mesh** or **Deformed Geometry** node. To see how these smoothing methods differ, let x and y be the spatial coordinates of the spatial frame, and let X and Y be the reference coordinates of the material frame.

- If Laplace smoothing is selected, the software introduces deformed mesh positions x and y as degrees of freedom in the model. In the static case, it solves the equation

$$\frac{\partial^2 x}{\partial X^2} + \frac{\partial^2 x}{\partial Y^2} = 0$$

  and in the transient case it solves the equation

$$\frac{\partial^2}{\partial X^2}\frac{\partial x}{\partial t} + \frac{\partial^2}{\partial Y^2}\frac{\partial x}{\partial t} = 0$$

  Similar equations hold for the y coordinate.

- If Winslow smoothing is selected, the software solves the equation

$$\frac{\partial^2 X}{\partial x^2} + \frac{\partial^2 X}{\partial y^2} = 0$$

  and does the same for Y. Equivalently, X and Y satisfy Laplace equations as functions of the x and y coordinates.

- The hyperelastic smoothing method searches for a minimum of a mesh deformation energy inspired by neo-Hookean materials:

$$W = \int_{\Omega} \frac{\mu}{2}(I_1 - 3) + \frac{\kappa}{2}(J - 1)^2 dV$$

  where μ and κ are artificial shear and bulk moduli, respectively, and the invariants J and I_1 are given by

$$J = \det(\nabla_X x)$$

$$I_1 = J^{-2/3}\mathrm{tr}((\nabla_X x)^T \nabla_X x)$$

- The Yeoh smoothing method is also inspired by hyperelastic materials, in this case the three-term Yeoh hyperelastic model, which is a generalization of a neo-Hookean material. It uses a strain energy of the form

$$W = \frac{1}{2}\int_{\Omega} C_1(I_1 - 3) + C_2(I_1 - 3)^2 + C_3(I_1 - 3)^3 + \kappa(J - 1)^2 dV$$

  where κ is an artificial bulk modulus, as above, while C_1, C_2, and C_3 are other artificial material properties. The values of C_1 and C_3 are by default 1 and 0, respectively, and can only be changed in the Equation View subnodes under a Free Deformation node. The value of C_2 controls the nonlinear stiffening of the artificial material under deformation. It is specified in the **Stiffening factor** field, with default value 100.

The Laplace smoothing is the cheapest option in terms of computations since it is linear and uses one equation for each coordinate direction, which are not coupled to each other. On the other hand, there is no mechanism in Laplace smoothing that prevents inversion of elements. Therefore, the method is most suitable for small deformations in a linear regime—for example, when computing the sensitivity of some quantity to small deformations around the initial shape.

The Winslow, Hyperelastic, and Yeoh smoothing methods are increasingly nonlinear and create a single coupled system of equations for all coordinate directions, which makes them more expensive to solve. They also share the theoretical property that continuous solutions to these equations always have positive volume everywhere. Unfortunately, this is not necessarily true for the discrete finite element solutions. In addition, a positive volume is not sufficient for maintaining element quality.

In compression, the three nonlinear methods show similar behavior, while in extension, the Winslow smoothing tends to allow elements to be stretched too far. The main difference between the simpler Hyperelastic method and the more advanced Yeoh model is that the latter responds to element distortion by sharply increasing the stiffness of distorted elements. This to some extent prevents further distortion in those regions and effectively acts to spread the mesh deformation more evenly over the domain, away from moving boundaries.

Yeoh smoothing generally produces the best results and allows the largest displacement of boundaries before mesh elements become inverted. However, because of its strong nonlinearity, it can cause convergence problems, in particular for the time dependent and segregated solvers.

Limitations of the ALE Method

The following limitations apply to the ALE method in general and therefore to the Moving Mesh and Deformed Geometry interfaces:

- The connectivity of the mesh remains unchanged during the mesh deformation, which means that topological changes in the geometry are not allowed.
- When the mesh deformation becomes large, the quality of the mesh created by the smoothing equations can deteriorate, and the solver might then run into convergence problems. Sometimes an **Inverted mesh element** warning displays in the **Progress** window for the solver, which means that a mesh element has (partially) warped inside-out. Sometimes, introducing extra boundaries with explicit deformation inside the domains can help. You can also generate a new mesh for the region covered by the deformed mesh and let the solver continue by deforming the new mesh; see the section Remeshing a Deformed Mesh. See also Tips for Modeling Using Deformed Meshes below.
- If you use a **Geometry shape order** larger than 1 in the Moving Mesh and Deformed Geometry interfaces, the mesh moving techniques often produce elements with distorted shapes. If there are warnings about inverted mesh elements, consider reducing the geometry shape order to 1. This, however, makes the geometry representation polygonal, which might affect accuracy. The measure of mesh quality does not capture these distorted shapes because it is computed from the positions of the corners of the mesh element (ignoring midside nodes, for instance).

Tips for Modeling Using Deformed Meshes

When working with a deformed mesh to move things around, the computational mesh gets deformed. If the deformations become too large, some mesh elements might get inverted. This means that the accuracy of the solution deteriorates and eventually the solvers diverge due to an ill-conditioned system. Here are some tips on how to keep the mesh under control:

- Try a different mesh. It is often preferable to start from a reasonably uniform mesh. One way to achieve this is to first select a coarse mesh in the predefined mesh size settings and then set a small maximum element size. Also, quad meshes and mapped meshes tend to perform better than triangles.
- Try a different smoothing type. Winslow smoothing is slightly slower, more memory consuming, and usually, but not always, more stable than Laplace smoothing. Hyperelastic and Yeoh smoothing sometimes work better than the other methods (in some fluid-structure interaction problems, for example). See Smoothing Methods for more information.
- If solving a time-dependent problem, try to solve the equations more accurately by reducing the absolute and relative tolerances for the time-dependent solver.
- Help the mesh deformation by sliding the boundary elements along with the movement of the mesh. This can be achieved by adding a prescribed deformation on the boundary that moves the boundary elements according to the deformation of some point in the model. Define a coupling operator under the **Definitions** node and use it to couple the deformation from the point to the boundary mesh.

Remeshing a Deformed Mesh

When the mesh deformation has become so large that the quality of the mesh is too bad, generate a new mesh for the deformed configuration and then continue the solution process. To do so, follow these steps:

1. Add a stop condition.
2. View the deformed mesh.
3. Copy the solution.
4. Create a Deformed Configuration ( ).
5. Remesh the deformed configuration.
6. Continue solving with the new mesh.

To make the Deformed Configuration mesh represent the deformed geometry, use a Moving Mesh (ALE) interface to model the mesh deformation or, for structural mechanics models, use the option in the study to include geometric nonlinearity (requires the Structural Mechanics Module, MEMS Module, or Acoustics Module). Also, some of the physics in the Corrosion Module and Electrodeposition Module include a deformed geometry.

- Deformed Configuration
- Solution (data set)

The following sections contain details about these steps and additional information.

ADDING A STOP CONDITION

Add a stop condition in the solver to make it stop when the mesh quality becomes too bad. If the Time-Dependent Solver is used, do this by right-clicking, for example, **Study 1>Solver Configurations>Solver 1>Time Dependent Solver 1** and selecting **Stop Condition** ( ) from the context menu. If the parametric stationary solver is used, right-click, for example, **Stationary Solver 1>Parametric 1** under **Solver 1** and select **Stop Condition** ( ) from the context menu.

In the table under **Stop Expressions** enter a Boolean expression (to stop when the expression is true) or an expression that makes the solver stop when the expression becomes negative. For example, enter `comp1.ale.I1isoMax>4` to stop before the maximum element distortion exceeds 4.

The following predefined variables are useful for defining a stop condition and for monitoring the mesh deformation: the maximum element distortion, `ale.I1isoMax`; the minimum relative element volume, `ale.relVolMin`; and the minimum mesh quality, `ale.minqual`. See Predefined Variables below.

In time-dependent simulations, you can use automatic remeshing instead of the stop condition. The software then creates new meshes when the mesh quality drops below the specified level. To do so, click the **Step 1: Time Dependent** node ( ) and then select the **Automatic remeshing** check box in the settings window's **Study Extensions** section. To use the same condition as for the stop condition above, select **Distortion** under **Condition type** and, enter `comp1.ale.I1isoMax` in the **Distortion expression** field and 4 in the **Stop when distortion exceeds** field in the settings window for the **Automatic Remeshing** ( ) node (under the **Time-Dependent Solver** node in the solver sequence).

VIEWING THE DEFORMED MESH

Use a Mesh plot in a 2D or 3D plot group to visualize the deformed mesh. The Mesh plot shows the element shapes, sizes and quality corresponding to the frame selected in the underlying data set. See Mesh (Plot) for details.

COPYING THE SOLUTION

To keep the first solution, right-click **Study 1>Solver Configurations>Solver 1** and select **Solution>Copy** ( ). The copied solution appears as a new solution **Copy 2**, for example.

CREATING A DEFORMED CONFIGURATION

Create a deformed configuration by right-clicking, for example, **Results>Data Sets>Solution 2** and selecting **Remesh Deformed Configuration** ( ). The deformed configuration appears as a new **Deformed Configuration** node ( ) under **Meshes**. The deformed configuration works as a new geometry but with restricted functionality. The settings window of the deformed configuration indicates which solution it was constructed from. Click the **Update** button to see the corresponding deformed configuration in the graphics.

REMESHING THE DEFORMED CONFIGURATION

Expanding a **Deformed Configuration** node ( ) shows that a new mesh sequence has been added beneath it. This mesh sequence contains a **Size** node ( ) and a **Reference** node ( ) only. The reference node refers to the original mesh sequence. This means that the new mesh sequence uses the same nodes as the original mesh sequence. Build the new mesh sequence by selecting **Build All** from its context menu or pressing F8. To make changes to the new mesh sequence before building it, right-click **Reference 1** and select **Expand** ( ). Then the nodes from the original mesh sequence are copied to the new mesh sequence. You can also add and remove nodes in the new mesh sequence. If several meshes are needed on the deformed configuration (for the multigrid solver, for example), you can add an additional mesh sequence by right-clicking **Deformed Configuration** ( ) and selecting **Mesh** ( ).

CONTINUE SOLVING WITH THE NEW MESH

1. In the settings window for the study step (for example, **Study 1>Step1: Time Dependent**), use the **Mesh** list under **Mesh Selection** to select the new mesh sequence (**Mesh 2**, for example).
2. Change the **Times** list under **Study Settings** or the **Parameter value list** under **Study Extensions** to include only the time or parameter corresponding to the deformed configuration plus the remaining times or parameters.

3 Change the initial value to be the last time or parameter of the previous solution. Do this by expanding Values of Dependent Variables in the study step settings. Select **Initial Values of Variables Solved For**, change **Method** to **Solution** and locate the previously created solution copy. Then select the appropriate time or parameter value in the list **Time** or **Parameter value**. Usually the **Automatic** alternative, which selects the last time or parameter value, suffices. Change the settings under **Values of Variables Not Solved For** similarly.

4 To solve for remaining times or parameters, right-click **Study 1** and select **Compute**.

REMESHING SEVERAL TIMES

You can remesh several times by iterating the above steps. For each of the solver runs you get a copy of the solution (**Copy 2**, **Copy 3**, **Copy 4**, and so on) and a corresponding data set (**Solution 2**, **Solution 3**, **Solution 4**, and so on). In the plot group, select one of these data sets for results analysis and visualization.

ALTERNATIVE PROCEDURE USING SEVERAL STUDIES

The above procedure uses a single study that is modified for each solver run. To recompute the whole sequence of runs, an alternative that uses one study for each run is better. To do that, add a new study after meshing each deformed configuration. The copy solution step is not needed in this case. If the settings are changed in the study or its solver sequence, make these changes also in the new study. For example, the stop condition has to be added.

Moving Mesh Interface

The **Moving Mesh (ale)** interface ( ), found under the **Mathematics>Deformed Mesh** branch ( ) when adding a physics interface, can be used to create models where the geometry, here represented by the mesh, changes shape due to some physical phenomena without material being removed or added. The difference between the Deformed Geometry and Moving Mesh interfaces is that the former defines a deformation of the material frame relative to the geometry frame, while the latter defines a displacement of the spatial frame relative to the material frame. The Moving Mesh interface can be used to study both stationary states and time-dependent deformations where the geometry changes its shape due to the dynamics of the problem. For example, it can be used for fluid domain deformations in fluid-structure interaction (FSI) or electrostatic domain deformations in MEMS.

When this physics interface is added, these default nodes are also added to the **Model Builder**—**Fixed Mesh** and **Prescribed Mesh Displacement** (the default boundary condition). Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions. You can also right-click **Moving Mesh** to select physics from the context menu.

Predefined Variables

The Moving Mesh interface includes the following predefined variables, which can be of interest, for example, to monitor the quality of the mesh and define a stop criterion for remeshing (see Adding a Stop Condition):

- Maximum element distortion, `ale.I1isoMax`, is measured as the maximum of the first invariant of the isochoric mesh strain tensor, ale.I1iso, over the moving mesh domains. This measure is zero for a mesh which is identical to the original material frame mesh except for a uniform scaling, and increases with increasing element distortion. Any element where `ale.I1iso` is about 2 or greater must usually be considered severely distorted. By plotting this quantity while solving, you can monitor how the mesh deforms and where it might become too distorted.
- The local relative element volume, `ale.relVol`, is a quantity that measures the local volumetric distortion of the elements. When this measure approaches zero in some part of the mesh, frame transformations become singular causing solvers to fail.
- The minimum relative element volume, `ale.relVolMin`, must be > 0, otherwise the mesh elements are inverted. A suitable stop criterion using this variable is that the minimum relative element volume must be larger than a small positive number.
- The maximum relative element volume, `ale.relVolMax`, is a positive scalar number that represents the maximum value of the relative element volume.
- The minimum mesh quality, `ale.minqual`, must be > 0; an acceptable mesh quality is typically larger than 0.1 (where the quality measure is a number between 0 and 1).

Sloshing Tank: model library path
COMSOL_Multiphysics/Fluid_Dynamics/sloshing_tank

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `ale`.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

> You do need to use a Moving Mesh interface in domains for which the displacements appear among the dependent variables, for example, where a Solid Mechanics interface is active.

FRAME SETTINGS

Specify the names of the spatial coordinates of the base frame for the physics interface—the material frame—in the **Material frame coordinates** fields. The defaults are the coordinates of the spatial frame in uppercase letters (**X**, **Y**, and **Z**). You can change the names in the fields for the **First**, **Second**, and **Third** coordinate. The field labels include the default spatial coordinate names in parentheses.

The **Geometry shape order** setting controls the order of polynomials—**1** (linear), **2** (quadratic—the default), **3** (cubic), **4** (quartic), or **5** (quintic, 2D only)—used for representing the geometry shape in the spatial frame. The same order is used for Lagrange shape functions defining the mesh position in domains where **Free displacement** has been activated.

FREE DEFORMATION SETTINGS

Select the smoothing type for freely deformed domains from the **Mesh smoothing type** list. Choose between Lagrange, Winslow, and hyperelastic smoothing. The default is Laplace smoothing.

> - Domain and Boundary Nodes in the Moving Mesh Interface
> - Smoothing Methods
> - Show More Physics Options

Domain and Boundary Nodes in the Moving Mesh Interface

The Moving Mesh Interface includes these domain and boundary nodes:

- Fixed Mesh
- Free Deformation
- Prescribed Deformation
- Prescribed Mesh Displacement
- Prescribed Mesh Velocity
- Prescribed Normal Mesh Velocity
- Zero Normal Mesh Velocity
- Zero Normal Mesh Displacement

Fixed Mesh

Use the **Fixed Mesh** node to specify that the selected domains remain at their reference material shape and do not move. This is the default.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

Prescribed Mesh Displacement

Use the **Prescribed Mesh Displacement** node on the boundary of domains with free deformation. The spatial frame in the adjacent domain moves in accordance with the displacement.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PRESCRIBED MESH DISPLACEMENT

Select the check box for each coordinate direction to prescribe the displacement. The default settings provide a fixed boundary (zero displacements in all directions).

For boundaries adjacent to domains where displacement variables are defined, for example domains where a Solid Mechanics interface is active, let these variables drive the mesh displacement by typing the component field names in the corresponding fields (for example, setting **dx** to `u`, **dy** to `v`, and **dz** to `w` in 3D).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Free Deformation

The **Free Deformation** node constrains the mesh displacement only by the boundary conditions on the surrounding boundaries. The displacement in the domain is obtained by solving a PDE.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

INITIAL DEFORMATION

Give initial values for mesh displacements, relative to the material frame, in the **Initial mesh displacement** fields.

Prescribed Deformation

Use the **Prescribed Deformation** node to define the deformation explicitly using expressions, or if you want the spatial coordinates to follow a deformation computed by, for example, a Solid Mechanics interface. (You can also achieve the latter effect by excluding the domains where the Solid Mechanics interface is defined from the domains where the Moving Mesh interface is active.)

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

PRESCRIBED MESH DISPLACEMENT

Specify expressions that define the deformation in the **Prescribed mesh displacement** fields (dx, dy, and dz for 3D models, for example) (SI unit: m). The default gives no mesh displacement.

Prescribed Mesh Velocity

Use the **Prescribed Mesh Velocity** node on the boundary of domains with free displacement to specify the velocity of the boundary. The spatial frame in the adjacent domains moves in accordance with the velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED MESH VELOCITY

Select the check box for each coordinate directory to prescribe a velocity. The default settings provide zero velocities in all directions.

CONSTRAINT SETTINGS

To display this section, click the **Show** button (👁) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Prescribed Normal Mesh Velocity

Use the **Prescribed Normal Mesh Velocity** node to specify the normal velocity of the boundary. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

Simulations using moving meshes, with a boundary moving in the normal direction, can sometimes need a stabilizing term to suppress the formation of local boundary segments of high curvature. This can be of particular importance in an automatic remeshing sequence, where the remeshing step might amplify local curvature artifacts.

The Moving Boundary Smoothing option smooths the normal mesh velocity of the Prescribed Normal Mesh Velocity node according to the following equation:

$$\frac{\partial \mathbf{X}}{\partial t}\cdot\mathbf{n} = v_0 + v_{\mathrm{mbs}}$$

where v_0 is the desired normal mesh velocity, and v_{mbs} is a smoothing velocity according to:

$$v_{\mathrm{mbs}} = \delta_{\mathrm{mbs}}|v_0|hH$$

Here δ_{mbs} is a moving boundary smoothing tuning parameter (unitless), h is the mesh element size (SI unit: m), and H the mean surface curvature (SI unit: 1/m), defined as:

$$H = -\frac{1}{2}\nabla_T\cdot\mathbf{n}$$

where ∇_T is the surface gradient operator.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

NORMAL MESH VELOCITY

Enter a value or expression for the **Normal mesh velocity** v_n (SI unit: m/s).

MOVING BOUNDARY SMOOTHING

By default, the **Enable moving boundary smoothing** check box is not selected. To enable boundary smoothing, click to select the check box, and then enter a value or expression for the **Moving boundary smoothing tuning parameter**, δ_mbs (unitless). The default value is 0.5.

Zero Normal Mesh Velocity

Use the **Zero Normal Mesh Velocity** node to set the normal velocity of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the physics interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

Zero Normal Mesh Displacement

Use the **Zero Normal Mesh Displacement** node to set the displacement in the normal direction of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set explicitly in the tangential direction. Note however that for curved boundaries, a zero normal mesh displacement implicitly results in a zero displacement also in the tangential direction. Only for flat parts of the undeformed mesh is the boundary free to move in the tangential direction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Deformed Geometry Interface

The **Deformed Geometry (dg)** interface (), found under the **Mathematics>Deformed Mesh** branch ( ) when adding a physics interface, can be used to study how physics changes when the geometry, here represented by the mesh, changes due to an externally imposed geometry change. The difference between the Deformed Geometry and Moving Mesh interfaces is that the former defines a deformation of the material frame relative to the geometry frame, while the latter defines a displacement of the spatial frame relative to the material frame. The Deformed Geometry interface can be used in cases where the original geometry model shrinks or grows by removal or addition of material. For example, it can be used for shape optimization (geometry shrinks and grows at different places simultaneously), corrosion (geometry shrinks), or electrodeposition (geometry grows).

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the identifier string must be unique. Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `dg`.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

!

The Deformed Geometry interface requires a domain selection which covers all domains in which some physics is active.

FRAME SETTINGS

Specify the spatial coordinates of the base frame for the Deformed Geometry interface— the geometry frame—in the **Geometry frame coordinates** fields. The default is upper case versions of the spatial coordinates followed by a lowercase g (for example, **Yg**). You can change the names in the fields for the **First**, **Second**, and **Third** coordinate. The field labels include the default spatial coordinate names in parentheses.

The **Geometry shape order** setting controls the order of polynomials used for representing the geometry shape in the material frame. The same order is used for Lagrange shape functions defining the mesh position in domains where **Free displacement** has been activated.

FREE DEFORMATION SETTINGS

Select the smoothing type for freely deformed domains. Choose between Lagrange, Winslow, and Hyperelastic smoothing.

For detailed information about selecting geometric entities (domains, boundaries, edges, and points), see Working with Geometric Entities

For more information about deformed meshes, see Moving Mesh Interface.

- Domain and Boundary Nodes for Deformed Geometry
- Smoothing Methods
- Show More Physics Options

Electrochemical Polishing: model library path **COMSOL_Multiphysics/Electromagnetics/electrochemical_polishing**

Domain and Boundary Nodes for Deformed Geometry

The Deformed Geometry Interface includes these domain and boundary nodes:

- Free Deformation
- Prescribed Deformation
- Fixed Mesh
- Prescribed Mesh Displacement
- Prescribed Mesh Velocity
- Prescribed Normal Mesh Velocity
- Zero Normal Mesh Velocity
- Zero Normal Mesh Displacement

Fixed Mesh

Use the **Fixed Mesh** node to specify that the selected domains retain their original shape as defined by the geometry and original mesh. This is the default.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

Prescribed Mesh Displacement

Add the **Prescribed Mesh Displacement** node on the boundaries of domains with free deformation. The material frame in the adjacent domain moves in accordance with the displacement.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED MESH DISPLACEMENT

Select the check box for each coordinate direction where you want to prescribe the displacement. The default settings provide a fixed boundary (zero displacements in all directions).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Free Deformation

The **Free Deformation** node constrains the mesh displacement only by the boundary conditions on the surrounding boundaries. The material frame displacement in the domain is obtained by solving a PDE.

Smoothing Methods

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL DEFORMATION

Give initial values for mesh displacements, relative to the material frame, in the **Initial mesh displacement** fields.

Prescribed Deformation

Use the **Prescribed Deformation** node to define the deformation of the material frame explicitly using expressions.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

PRESCRIBED MESH DISPLACEMENT

Specify expressions that define the deformation in the **Prescribed mesh displacement** fields. Select the check box to enable the prescribed mesh displacement in the directions to use such a displacement condition. Use one expression per spatial coordinate.

Prescribed Mesh Velocity

Use the **Prescribed Mesh Velocity** node on the boundary of domains with free displacement. The material frame in the adjacent domains moves in accordance with the velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED VELOCITY

Select the check box for each coordinate directory to prescribe a velocity. The default settings provide zero velocities in all directions.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Prescribed Normal Mesh Velocity

Use the **Prescribed Normal Mesh Velocity** node to specify the normal velocity of the boundary. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

Simulations using moving meshes, with a boundary moving in the normal direction, can sometimes need a stabilizing term to suppress the formation of local boundary segments of high curvature. This can be of particular importance in an automatic remeshing sequence, where the remeshing step might amplify local curvature artifacts.

The Moving Boundary Smoothing option smooths the normal mesh velocity of the Prescribed Normal Mesh Velocity node according to the following equation:

$$\frac{\partial \mathbf{X}}{\partial t} \cdot \mathbf{n} = v_0 + v_{\mathrm{mbs}}$$

where v_0 is the desired normal mesh velocity, and v_{mbs} is a smoothing velocity according to:

$$v_{\mathrm{mbs}} = \delta_{\mathrm{mbs}} |v_0| hH$$

Here δ_{mbs} is a moving boundary smoothing tuning parameter (unitless), h is the mesh element size (SI unit: m), and H the mean surface curvature (SI unit: 1/m), defined as:

$$H = -\frac{1}{2} \nabla_T \cdot \mathbf{n}$$

where ∇_T is the surface gradient operator.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

NORMAL MESH VELOCITY

Enter a value or expression for the **Normal mesh velocity** v_{n} (SI unit: m/s).

MOVING BOUNDARY SMOOTHING

By default, the **Enable moving boundary smoothing** check box is not selected. To enable boundary smoothing, click to select the check box, and then enter a value or expression for the **Moving boundary smoothing tuning parameter**, δ_{mbs} (unitless). The default value is 0.5.

Zero Normal Mesh Velocity

Use the **Zero Normal Mesh Velocity** node to set the normal velocity of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Zero Normal Mesh Displacement

Use the **Zero Normal Mesh Displacement** node to set the displacement in the normal direction of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set explicitly in the tangential direction. Note however that for curved boundaries, a zero normal mesh displacement implicitly results in a zero displacement also in the tangential direction. Only for flat parts of the undeformed geometry is the boundary free to move in the tangential direction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

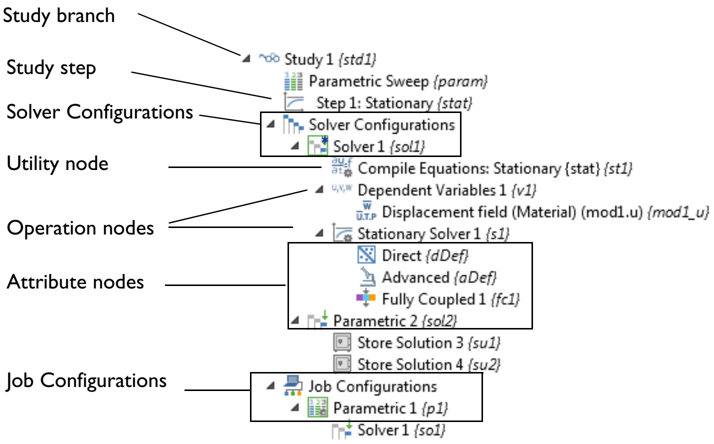
19

Studies and Solvers

This chapter describes the study types and solvers available in COMSOL Multiphysics®.

Introduction to Solvers and Studies

The process of solving a problem in COMSOL Multiphysics is a hierarchy. The Study node ( ) is the coarsest level (the top level). It contains the least amount of detail and defines a Study branch (see Branches and Subbranches in the Tree Structure).



Study 1 *{std1}*
Parametric Sweep *{param}*
Step 1: Stationary *{stat}*
Solver Configurations
Solver 1 *{sol1}*
Compile Equations: Stationary {stat} *{st1}*
Dependent Variables 1 *{v1}*
Displacement field (Material) (mod1.u) *{mod1_u}*
Stationary Solver 1 *{s1}*
Direct *{dDef}*
Advanced *{aDef}*
Fully Coupled 1 *{fc1}*
Parametric 2 *{sol2}*
Store Solution 3 *{su1}*
Store Solution 4 *{su2}*
Job Configurations
Parametric 1 *{p1}*
Solver 1 *{so1}*

Figure 19-1: An example of the hierarchy under the Study node. This is the completed study and solver sequence for the Diagonal Mounting Detail of a Communication Mast model from the COMSOL Multiphysics model library. Some hierarchical categories for this sequence are shown.

Add a Study and Study Steps

While Creating a New Model, you can add any of the predefined Study Types. At any time you can also use The Add Study Window. However you choose to add a study, a study node is added to The Model Builder including a corresponding *study step* (for example, **Stationary** in Figure 19-1), and in some cases, additional study steps. The study step represents the next level of detail.

Define Study Steps and Create Solver Configurations

Most study steps are used to control the form of the equations, what physics are included in the computation, and what mesh is used. A study step settings window has a Physics and Variables Selection section where inclusion and exclusion of physics interfaces and variables can be adjusted and set. There are also Common Study Step Settings for many of the study features added to a sequence.

Study steps correspond to part of a *solver configuration* (solver sequence), which is the next level of detail. There are also study steps that are used for cluster computing, for example, which correspond to part of the Job Configurations.

Solver Configurations contain nodes that define variables to solve for, the solvers and settings, and additional sequence nodes for storing the solution, for example (see Figure 19-1). The solvers also have nodes that can control the solver settings in detail. Knowing The Relationship Between Study Steps and Solver Configurations is useful to help define and edit the settings before computing a solution.

Compute the Solution

Once the studies are added and defined, the simplest option to compute the solution is to right-click the **Study** node for a predefined study type and select **Compute** (=). This generates the default solver configuration for the corresponding study steps and computes the solution. There are a variety of techniques you can use while Computing a Solution, including many custom adjustments.

CONTROL AND CUSTOMIZE SOLVER SETTINGS

The settings can also be controlled at any level of detail. For example, you can add individual study steps when there is not a predefined study type that corresponds to the simulations you are interested in doing. Also, by changing the settings in the solver configuration you can, for example, control the desired tolerance for the error in the solution or which time-integration method or linear solver to use.

Solver Operation, Attribute, and Utility Nodes

There are different groups of feature types you can use to customize and fine tune the model. There are three classes of features that are subnodes to a Solver node. See Figure 19-1 for examples:

- *Operation nodes* (typically solvers) produce solutions as output. In particular, the output from the last executed operation node is available for results analysis and visualization. See Solution Operation Nodes and Solvers.
- *Attribute nodes* hold properties that control the behavior of operation features. See Solution Attribute Nodes.
- *Utility nodes* handle special types of operations. Applicable solution utility nodes are available from the **Solver>Other** submenu. See Solution Utility Nodes.

Some of the settings in subnodes are synchronized with the corresponding Study setting. They are unavailable by default and can only be controlled from the subnode by changing the **Defined by study step** setting to **User defined**.

There are also Study Extension Steps (Parametric Sweep and Optimization) and categories of Advanced Study Extension Steps (Parametric, Batch, and Cluster Computing) for additional settings customization.

For some modules, more settings are available with respect to Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.

About Solver Commands and Solution Object Data in the *COMSOL API Reference Manual*

The Add Study Window

The **Add Study** window is similar to the **Select Study Type** page accessed through The Model Wizard. It has the same studies available and is a quick way to add a study (or studies) to models. You can have more than one study (each generating one or more solutions) for different scenarios using the same geometry and physics. The predefined study types correspond to the most commonly performed simulations for different physics. However, sometimes you might want to do other investigations. For example, to solve a stationary problem for a physical quantity and use that solution as input to a time-dependent simulation for another physical quantity. For example, add a **Stationary** study step ( ), followed by a **Time Dependent** study step ( ) and then for each study step, choose the physics to include.

To open the **Add Study** window, right-click the **Root** node and choose **Add Study** . or use one of the following alternatives:

Win

- From the **Home** ribbon, click **Add Study** or select **More Windows>Add Study** .
- From the **Study** toolbar, click **Add Study** .

Mac

Linux

- On the Model Toolbar, click **Add Study** .
- On the **Study** toolbar click **Add Study** .
- Select **Windows>Add Study** .

The **Add Study** toolbar button is a toggle button: Click it again to close the **Add Study** window.

TO ADD A STUDY TO A MODEL ROOT NODE

1 In the **Add Study** window, from the branches under **Studies**, select the type of study to perform. The available options depend on the set of physics interfaces included in the model. Some study types are applicable to all physics for which you choose to solve, while others are not, but in some way all are available. Select the study type from one of the following branches:

- **Preset Studies**—Study types suggested by a single physics interface if only one has been chosen.
- **Preset Studies for Selected Physics**—Study types applicable to all physics interfaces that you have chosen to solve for.
- **Custom Studies**—This branch contains study types for which not all physics solved for can generate suitable equations.
- **Custom Studies>Preset Studies for Some Physics**—The study types recognized by some, but not all, of the physics being solved for.
- **Custom Studies>Other studies**—Any fundamental study types (**Stationary**, **Time Dependent**, **Eigenfrequency**, **Eigenvalue** and **Frequency Domain**) that are not applicable to any of the physics being solved for. There is also an empty study type.

2 When there is already a physics interface in the model, the existing **Physics** are listed under **Physics in study**. Physics interfaces that are included appear with a check mark (✓) in the **Solve** column. Click in the row and column to change the check mark to an (✗).

3 Click the **Add Study** button. The study is added under the **Study** node in the **Model Builder**

- Creating a New Model
- Study Types

Study

A **Study** node ( ) holds the all the nodes that define how to solve a model (see Figure 19-1). These nodes are divided into three broad categories:

- Study steps, which determines overall settings suitable for a certain study type. The study steps added are based on the Study Types chosen.
- Solver Configurations, which contains the solvers and related configurations for dependent variables to solve for, intermediate storage of solutions, and specific solver settings.
- Job Configurations, contains all jobs defined for a study (distributed parametric jobs, batch jobs, and cluster computing).

The main **Study** node has one section:

STUDY SETTINGS

The **Generate default plots** check box is selected by default so that plot groups with suitable default plots for the physics in the study are generated automatically when computing the solution. Clear this check box if you do not want any default plots.

The **Generate convergence plots** check box is selected or cleared by default based on the setting for generating convergence plots in the **Preferences** dialog box. Clear this check box if you do not want convergence plots to be generated during the solution process.

- The Add Study Window
- Convergence Plots
- The Relationship Between Study Steps and Solver Configurations

Solver Configurations

The **Solver Configurations** node ( ) contains all solver configurations defined for a study (see Figure 19-1). It displays if it has content or, to make the node available in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

A model is solved by computing a *solver configuration*—a scheme for computing a solution. Loosely speaking a solver configuration consists of one or more **Solver** nodes ( ), and each Solver node consists of a sequence of subnodes specifying how to compute the solution. Typically, such a solver configuration contains information about which physics and geometry to use, which variables to solve for, and which solvers to use for the type of study to perform. You can also solve a model by computing a study; this defines a sequence of solver configurations and, in some cases, a sequence of Job Configurations.

Right-click the node to choose one of these options from the context menu—**Show Default Solver**, **Clear Solutions**, **Delete Solvers**, or **Create Custom Solver**. You can also edit and run or compute solver configurations (see Computing a Solution).

SHOW DEFAULT SOLVER

To display the solver that corresponds to the study steps in a study and the current physics settings, right-click the main **Study** node ( ) (or the Solver Configurations or Job Configurations nodes) and select **Show Default Solver** ( ).

CLEAR SOLUTIONS OR DELETE SOLVERS

Select **Clear Solutions** ( ) to remove all solutions in the study. See Clear or Delete a Mesh or Solution.

Select **Delete Solvers**( ) to delete all solvers under the **Solver Configurations** node.

CREATE A CUSTOM SOLVER

From the main menu, click the **Show** button ( ) and select **Advanced Study Options** and then right-click the **Solver Configurations** node and choose **Create Custom Solver** ( ). This add a **Solver** node without any added solver settings or other nodes.

- Computing the Initial Values
- The Relationship Between Study Steps and Solver Configurations
- Saving and Opening Recovery Files

The Relationship Between Study Steps and Solver Configurations

Most studies and study steps correspond to part of a solver configuration that includes a solver for the specific problem, as listed in Table 19-1.

- Study
- Solver Configurations

TABLE 19-1: THE RELATIONSHIP BETWEEN COMMON STUDY TYPES AND SOLVERS

STUDY TYPE	DESCRIPTION	CORRESPONDS TO SOLVER
Stationary	Generates equations without time derivatives.	Stationary Solver. A parametric continuation solver can also be created by selecting an option on the study settings window. Also see About the Stationary Solver and About the Parametric Solver.
Time Dependent	Generates equations for transient (time-dependent) simulations.	Time-Dependent Solver. Also see About the Time-Dependent Solver.
Time Discrete	Use it for performing time-dependent analysis using the projection method.	Time Discrete Solver. Also see About the Time Discrete Solver.
Time Dependent	Use the Time Explicit Solver to solve a time-dependent problem using an explicit time-stepping scheme.	Time Explicit Solver. Also see The Time Explicit Solver Algorithms.
Eigenvalue	Generates equations formulated for computing eigenvalues and eigenfunctions.	Eigenvalue Solver. Also see The Eigenvalue Solver Algorithm.
Eigenfrequency	Similar to an Eigenvalue study step.	Eigenvalue Solver (set to transform eigenvalues to eigenfrequencies). Also see The Eigenvalue Solver Algorithm.
Frequency Domain	Generates stationary equations that are used for frequency sweeps.	It corresponds to a stationary parametric solver that is preset to linearize the equations (Stationary Solver with a Parametric attribute). By selecting the Use asymptotic waveform evaluation check box, this study step corresponds to an AWE Solver.
Time-Dependent Modal	Generates equations for time-dependent modal analysis.	Modal Solver (with Study type set to Time dependent). Also see The Modal Solver Algorithm.
Frequency Domain Modal	Generates equations for modal analysis in the frequency domain.	Modal Solver (with Study type set to Frequency domain). Also see The Modal Solver Algorithm.

There are some study steps that do not generate equations and can only be used in combination with other study steps. These *study extension steps* do not correspond directly to any part of a solver configuration. Instead, they either correspond to a part of the job configuration or modify the behavior of another study step.

Study Extension Steps

A Parametric Sweep is used to formulate a sequence of problems that arise when you vary some parameters in the model. The problem at a fixed parameter value is defined by the rest of the study steps in the study. It generates a Parametric (Job Configuration), unless the problem and parameters are such that the parametric sweep can be realized through a Stationary Solver with a Parametric node, in which case such a solver is generated in the solver configuration.

The parametric sweep can include multiple independent parameters directly, but you can also add more than one Parametric Sweep node to create nested parametric sweeps. In the **Study** branch, indentations of the node names indicate that the parametric sweeps are nested.

The Optimization study step is used to solve PDE-constrained optimization problems. This study step allows direct definition of objective functions and selection of model parameters, including parameters that control the geometry, for optimization. It also provides detailed control over solvers and contributions to an optimization problem defined by an Optimization interface. This study type requires an Optimization Module license.

Advanced Study Extension Steps

Batch and Batch Sweep

A Batch creates a job that can be run without the graphical user interface and which stores the solution on disk. It generates a Batch (Job Configuration).

A Batch Sweep is used to formulate a sequence of problems that arise when you vary some parameter in the model. Each parameter tuple generates a batch job that runs the model with the given tuple. The results are stored on file and updated into the model. It generates a Batch (Job Configuration) and a Parametric (Job Configuration).

Cluster Computing and Cluster Sweep

A Cluster Computing study is used to solve the problem on a distributed-memory computer architecture. It generates a Cluster Computing (Job Configuration) and a Batch (Job Configuration).

A Cluster Sweep is used to formulate a sequence of problems that arise when you vary some parameter in the model. The program computes the solution for each parameter on a distributed-memory computer architecture. The results are stored on file and updated into the model. It generates a Cluster Computing (Job Configuration), Batch (Job Configuration), and (if applicable) Parametric (Job Configuration).

Multigrid Level

Multigrid Level can be added as a subnode to other study step nodes to describe a geometric multigrid level used by the study.

Sensitivity

The Sensitivity study step specifies objective functions and controls variables with respect to which sensitivity is computed. Global scalar objective functions can be specified directly in the study step, and model parameters selected as control variables. In addition, the study step provides control over the sensitivity solver method and contributions to the sensitivity problem defined with The Sensitivity Interface or Optimization interface (this study type requires an Optimization Module license).

Study Types

To add a study or study step see these topics:

- Creating a New Model
- The Add Study Window
- The Model Wizard

The main study types are listed in Table 19-2. Some studies require add-on modules, and these are listed in Table 19-3 (in that list, the submenu under **Study Steps** in the **Study** branches appears in parentheses for the study types that are available as individual study steps).

Also see Study Extension Steps and Advanced Study Extension Steps for additional information about some study steps that do not generate equations and which can only be used in combination with other study steps. In Table 19-2, these are listed as supplemental studies and study steps.

TABLE 19-2: STANDARD STUDY TYPES AND STUDY EXTENSIONS

ICON	STUDY TYPE	BRIEF DESCRIPTION
	Empty Study	An Empty Study creates a Study node with no study steps.
	Custom Studies	This branch contains study types for which the selected physics are not automatically adapted. Instead, these physics interfaces have to be adapted manually using the Model Builder.
	Preset Studies/Preset Studies for Selected Physics	If you have added a single physics interface, you find suggested studies under this node. If you have added multiple physics interfaces, you find studies applicable to all physics interfaces that you have added.
	Preset Studies for Some Physics	Under this node, studies applicable to all physics that you have chosen to solve for appear.
Stationary Submenu		
	Stationary	For a stationary or steady-state situation where you can use a stationary solver. This study type is also used for optimization problems that are constrained with a stationary PDE. Adds a Stationary study step. You can also choose to create a parametric continuation solver.
Time Dependent Submenu		
	Time Dependent	For a time dependent or transient simulation using a Time-Dependent Solver for computing the solution over time. This study type is also used for optimization problems that are constrained with a time-dependent PDE. Adds a Time Dependent study step.
	Time Discrete	Use it to perform time-dependent analysis using the projection method. Adds a Time Dependent study step.
	Time-Dependent Modal	This study is for analyzing time-dependent wave problems using a modal solver, the Time-Dependent Modal study adds an Eigenfrequency study step followed by a Time-Dependent Modal study step.

TABLE 19-2: STANDARD STUDY TYPES AND STUDY EXTENSIONS

ICON	STUDY TYPE	BRIEF DESCRIPTION
Eigenfrequency Submenu		
	Eigenfrequency	This study is similar to an Eigenvalue study but computes the eigenfrequencies instead of the eigenvalues. Adds an Eigenfrequency study step.
	Eigenvalue	This study uses a formulation to compute eigenvalues and eigenmodes using an eigenvalue solver. Adds an Eigenvalue study step.
Frequency Domain Submenu		
	Frequency Domain	For a study in the frequency domain such as wave equations or frequency response analysis. Adds a Frequency Domain study step.
	Frequency Domain Modal	The Frequency-Domain Modal study is for analyzing wave problems in the frequency domain using a modal solver, it adds an Eigenfrequency study step followed by a Frequency Domain Modal study step.
Study Extensions		
	Batch	Use to start a COMSOL Multiphysics batch process that solves the current study on your computer.
	Batch Sweep	Use to find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest.
	Cluster Computing	Use when you want to submit COMSOL Multiphysics batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster.
	Cluster Sweep	Use to find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest.
	Modal Reduced Order Model	To export the reduced-order model matrices for time-dependent wave problems using a modal solver, this study adds an Eigenfrequency study step followed by a Modal Reduced Order Model study step.
	Multigrid Level	This can be added as a subnode to other study step nodes to describe a geometric multigrid level used by the study.
	Parametric Sweep	Use to find the solution to a sequence of stationary or time-dependent problems that arise when you vary some parameters of interest. Add to a study to perform a parametric variation on other studies.
	Sensitivity	Use this to add a sensitivity analysis to the study. Using a Sensitivity study node you can add sensitivity functions at the study level and use model parameters as global control variables.

TABLE 19-3: STUDIES AND STUDY STEPS AVAILABLE WITH ADD-ON MODULES

ICON	STUDY TYPE	DESCRIPTION	REQUIRED MODULE
Chemical Applications			

TABLE 19-3: STUDIES AND STUDY STEPS AVAILABLE WITH ADD-ON MODULES

ICON	STUDY TYPE	DESCRIPTION	REQUIRED MODULE
	AC Impedance Stationary	Two study steps solve for a stationary problem and a harmonic perturbation in the frequency domain of the stationary solution.	Batteries & Fuel Cells Module, Corrosion Module, Electrochemistry Module, or Electrodeposition Module.
	AC Impedance Time Dependent	Two study steps solve for a time-dependent problem and a harmonic perturbation in the frequency domain of the solution at the last time step.	
	Cyclic Voltammetry (under Time Dependent)	Uses the Electroanalysis interface to perform transient simulations of voltammetry experiments.	
	Stationary Plug Flow (under Stationary)	Used with the plug flow reactor type with a Reaction Engineering interface and solves for the molar flow rate as function of reactor volume.	Chemical Reaction Engineering Module
	Time-Dependent with Initialization	Used to perform transient simulations of electrochemical cells. Solves for the electrode and electrolyte potentials as well as all global ODE dependent variables. It performs a transient simulation for all dependent variables in the model, using the result of the first study step as initial values.	Corrosion Module or Electrodeposition Module
	Time-Dependent with Initialization, Fixed Geometry	Use this study to exclude geometry deformation effects from a model. The study is similar to the Time-Dependent with Initialization study.	Corrosion Module or Electrodeposition Module
Electrical Applications			
	Boundary Mode Analysis	Combines a mode analysis on a port (boundary) with a frequency domain study for the full geometry. Adds a Boundary Mode Analysis study step followed by a Frequency Domain study step.	RF Module
	Coil Current Calculation	Use it to solve an eigenvalue problem for the current flow in a Multi-Turn Coil Domain node that gives the current density likely produced by a bundle of conductive wires.	AC/DC Module
	Frequency-Stationary	This is a special case of a Stationary study and is available with the Induction Heating and Microwave Heating interfaces.	AC/DC Module, RF Module, Wave Optics Module

TABLE 19-3: STUDIES AND STUDY STEPS AVAILABLE WITH ADD-ON MODULES

ICON	STUDY TYPE	DESCRIPTION	REQUIRED MODULE
	Frequency-Transient	Compute electromagnetic fields in the frequency domain and the temperature (or electron temperature) in the time domain. Available with the Induction Heating, Microwave Heating, Inductively Coupled Plasma, and Microwave Plasma interfaces.	AC/DC Module, Plasma Module, RF Module, or Wave Optics Module
	Mean Energies (under Frequency Domain)	Use to enter an array of values for the mean electron energy. Available with the Boltzmann Equation, Two-Term Approximation interface.	Plasma Module
	Reduced Electric Fields (under Frequency Domain)	Use to sweep through a range of reduced electric fields. Available with the Boltzmann Equation, Two-Term Approximation interface.	Plasma Module
Fluid Applications			
	Frozen Rotor	This is a special case of a Stationary study. The frozen rotor approach assumes that the flow in the rotating domain, expressed in the rotating coordinate system is fully developed. Available with the Rotating Machinery, Laminar Flow and Turbulent Flow interfaces.	CFD Module or Mixer Module
	Frozen Rotor with Initialization	For flow in rotating machinery where the topology of the geometry does not change with rotation. You can also use it to compute the initial conditions for time-dependent simulations of flow in rotating machinery.	CFD Module or Mixer Module
	Stationary with Initialization	For stationary two-phase flow models that require an initialization of a level set or phase field function. Also for turbulent flow models.	CFD Module or Heat Transfer Module
	Transient with Initialization	For time-dependent two-phase flow models that require an initialization of a level set function or phase field function. Also for turbulent flow models.	CFD Module, Heat Transfer Module, or Microfluidics Module
Mechanical Applications			
	Linear Buckling	Use this study type for a structural model to solve for the critical load factor using an eigenvalue solver.	Structural Mechanics Module or MEMS Module
	Prestressed Analysis, Eigenfrequency	For computing eigenfrequencies that are influenced by a prior static load.	Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module
	Prestressed Analysis, Frequency Domain	For computing the response to harmonic loads fluctuating around a static preload.	

TABLE 19-3: STUDIES AND STUDY STEPS AVAILABLE WITH ADD-ON MODULES

ICON	STUDY TYPE	DESCRIPTION	REQUIRED MODULE
	Stationary, One-Way Coupled	Two study steps solve for the fluid flow variables and for the solid deformation. Available with the Fluid-Structure Interaction interface.	MEMS Module or the Structural Mechanics Module
	Time Dependent, One-Way Coupled	Two study steps solve for the fluid flow variables and for the solid deformation in the time domain. Available with the Fluid-Structure Interaction interface.	
	Stationary, One-Way Coupled with Initialization	Three study steps solve for the distance to the closest wall, for the fluid flow variables, and for the solid deformation. Available with the Fluid-Structure Interaction interface.	MEMS Module or the Structural Mechanics Module, plus the CFD Module
	Transient, One-Way Coupled with Initialization	Three study steps solve for the distance to the closest wall, for the fluid flow variables, and for the solid deformation in the time domain. Available with the Fluid-Structure Interaction interface.	
Multipurpose Applications			
	Frequency-Domain, Perturbation (under Frequency Domain)	Used for studies of small oscillations about a bias solution. Two study steps compute the stationary (or bias) solution followed by a perturbed solution of the linearized problem around the linearization point (or bias point) computed in the first step.	Batteries & Fuel Cells Module, Electrodeposition Module, AC/DC Module, or MEMS Module
	Mode Analysis	Computes the modes for an acoustic or electromagnetic wave using an eigenvalue solver.	Acoustics Module or RF Module
	Optimization	Used to solve PDE-constrained optimization problems. This study step allows direct definition of objective functions and selection of model parameters, including parameters that control the geometry, for optimization. It also provides detailed control over solvers and contributions to an optimization problem defined by an Optimization interface.	Optimization Module

TABLE 19-3: STUDIES AND STUDY STEPS AVAILABLE WITH ADD-ON MODULES

ICON	STUDY TYPE	DESCRIPTION	REQUIRED MODULE
	Particle Trajectories (under Time Dependent)	This study has the same settings as the Time Dependent study step except that by default, only the Particle Tracing Module's interfaces for particle tracing are active in the physics selection.	Particle Tracing Module
	Small-Signal Analysis, Frequency Domain	For perturbed frequency domain studies of small oscillation about a bias solution. The study creates two study steps that solve for a stationary problem, and for a harmonic perturbation in the frequency domain of the stationary solution.	AC/DC Module, MEMS Module, or Semiconductor Module

Common Study Step Settings

The study steps form a solver configuration that computes the solutions for the study. The study step nodes' settings windows contain the following sections (in addition to specific study settings for each type of study step):

STUDY SETTINGS

Include Geometric Nonlinearity Check Box

If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module (including any add-on modules such as the Nonlinear Structural Materials, Fatigue, or Geomechanics Modules) and your model involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' manuals.

RESULTS WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving in the **Graphics** window. Then select what to plot from the **Plot group** list and, for time dependent simulations, at which time steps to update the plot: the output times or the time steps taken by the solver. The software plots the data set of the selected plot group as soon as the results become available. You can also control which probes to tabulate and plot the values from. The default is to tabulate and plot the values from all probes in the **Table** window and a **Probe Plot** window.

Use the Probes list to select any probes to evaluate. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** ( ), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving.

You can use probes to tabulate values of interest during a large parametric simulation, for example. It can then be possible to keep only the last solution in memory during the parametric sweep, which potentially can significantly reduce memory requirements and the simulation time.

PHYSICS AND VARIABLES SELECTION

See Physics and Variables Selection for detailed information about this section. Control and specify different cases where the physics to solve for is varied, or, for various analysis cases, which variables and physics features (for example, boundary conditions and sources) to use. The default is to solve for all physics that are compatible with the study type.

VALUES OF DEPENDENT VARIABLES

When you have physics in a study step that you do not solve for but that provide degrees of freedom, you can specify how COMSOL handles the values of such degrees of freedom (dependent variables).

The settings in this section determine how the solver handles dependent variables that you do not solve for. This is applicable in, for example, a solver configuration where you only solve for a subset of the dependent variables in each step.

By default, COMSOL determines these values heuristically depending on the model as, for example, the specified initial values or a solution from an earlier study step. To specify the initial values of the dependent variables that you solve for, select the **Initial values of variables solved for** check box. To specify the values of dependent variables that you do not solve for, select the **Values of variables not solved for** check box. Then use the **Method** list to specify how to compute the values of variables not solved for. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object (a solution from a study step).

Use the **Study** list to specify what study to use if **Method** has been set to **Solution**:

- Select **Zero solution** to initialize all variables to zero.
- Select any other available study to use it as initial value.

Depending on the study type, you can choose different solutions from a list underneath the **Study** list:

- For a Stationary study, from the **Selection** list, select **Automatic** (the default) to use the last (typically the only) solution, select **First** to use the first (typically the only) solution, select **Last** to use the last (typically the only) solution, select **All** to use all (typically just one) solutions from that study, select **Manual** to use a specific solution number that you specify, or select **1** to use the first (typically the only) solution. If you use a parametric continuation of the stationary study, there can be additional solutions to choose from.
- For a Time Dependent study, from the **Time** list, select **Automatic** (the default) to use the solution for the last time, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Interpolated** to specify a time in the text field that opens and use the interpolated solution at that time, select **Manual** to use a specific solution number that you specify, or select one of the output times to use the solution at that time. For all the options in the **Time** list (except **All**) one solution is used throughout the whole simulation. This solution is computed once before the simulation. When you select **All**, an interpolation is done internally for time-dependent simulations.
- For an Eigenvalue study, from the **Selection** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the eigenvalues to use the corresponding eigensolution.
- For a parametric or Frequency Domain study, from the **Parameter value** list, select **Automatic** (the default) to use the first parameter value set or frequency, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the parameter value sets or frequencies to use the corresponding solution.

MESH SELECTION

Specify—for each geometry—which mesh to use for the study step. For each geometry listed in the **Geometry** column, select a mesh from the list of meshes in the **Mesh** column. Each list of meshes contains the meshes defined for the geometry that you find on the same row.

STUDY EXTENSIONS

These are extensions to the study's main solver such as adaptive mesh refinement and automatic remeshing. The options vary depending on the study type.

Auxiliary Sweep

Select the **Auxiliary sweep** check box to enable an auxiliary parameter sweep, which corresponds to a Parametric solver attribute node. For each set of parameter values, the chosen **Sweep type** is solved for. This is available for Stationary, Time Dependent, and Frequency Domain studies.

Select a **Sweep type** to specify the type of sweep to perform.

- **Specified combinations** (the default) solves for a number of given combinations of values as given for each parameter in the list. The parameter lists are combined in the order given, that is, the first combination contains the first value in each list, the second combination contains all second values, and so on.
- **All combinations** solves for all combination of values of values, that is, all values for each parameter are combined with all values for the other parameters. Using all combinations can lead to a very large number of solutions (equal to the product of the lengths of the parameter lists).

In the table, specify the **Parameter names** and **Parameter value list** for the parametric solver. Click the **Add** button (+) to add a row to the table. When you click in the **Parameter value list** column to define the parameter values, click the **Range** button ( ) to define a range of parameter values.

If you choose **Specified combinations**, the list of values must have equal length.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. You can use the **Load from File** button (📂) to browse to such a text file. The read names and values are appended to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values and a space separating the values.

Click the **Save to File** button (💾) to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®).

For the Stationary or Frequency Domain study, select an option from the **Run continuation for** list—**No parameter** or one of the parameters given in the list.

Select an option from the **Reuse solution for previous step** list.

Select:

- **No** (the default for a Stationary study) to reset the solution to the initial values before each step or continuation sweep.

- **Yes** to always use the converged solution from the previous step, or the last solution from the previous continuation sweep (that is, never reset the solution).
- **Auto** (the default for a Frequency Domain study) to normally use the converged solution from the previous step or sweep. However, when multiple parameters are used, the solution from the first step of each parameter list is always used for the first step of the next list.

The difference between the three options is shown in Figure 19-2 for a 3x4 two-parameter sweep using the different choices for **Reuse solution for previous step** without continuation:

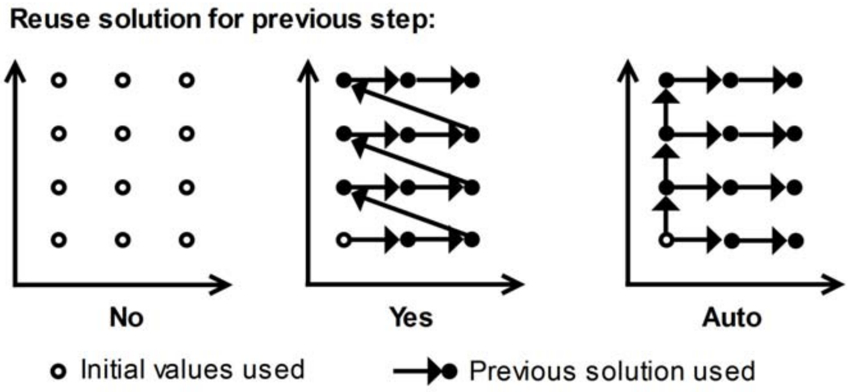


Figure 19-2: The difference between the three options for a two parameter sweep without continuation.

When continuation is enabled by setting **Run continuation for** to one of the parameters, the converged solutions are always reused for the steps along the continuation sweep in this parameter. The setting for **Reuse solution for previous step** then determines how the solutions are reused between multiple continuation sweeps, if there are additional parameters to sweep over, as shown in Figure 19-3.

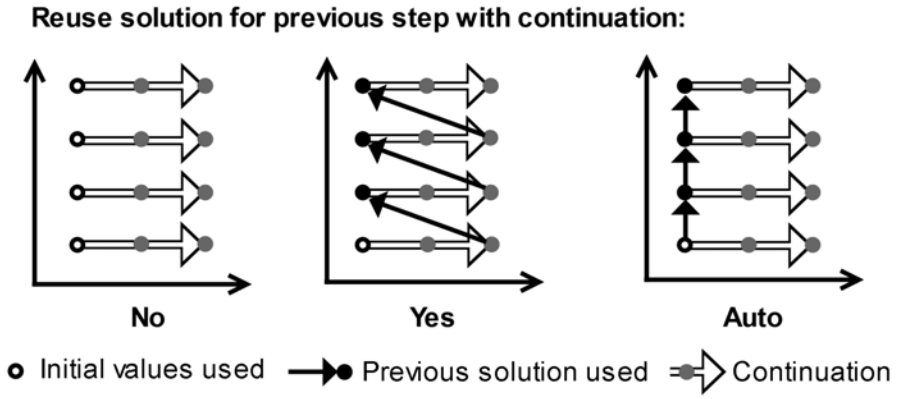


Figure 19-3: The difference between the three options for a two parameter sweep with continuation.

For the Frequency Domain study, the auxiliary sweep is merged with the frequency sweep into a multiparameter sweep with the frequency as the parameter at the innermost level.

See About the Parametric Solver.

Buoyancy Flow in Free Fluids: model library path **COMSOL_Multiphysics/Fluid_Dynamics/buoyancy_free**

With the AC/DC Module, see Small-Signal Analysis of an Inductor, model library path **ACDC_Module/Inductive_Devices_and_Coils/small_signal_analysis_of_inductor**

Adaptive Mesh Refinement

Select the **Adaptive mesh refinement** check box if you want to use adaptive mesh refinement. Select the geometry to use for the adaption from the **Adaption in geometry** list. Click the **Go to Source** button ( ) to move to the settings window for the geometry. See Adaptive Mesh Refinement for information about the subnode, its settings, and the algorithm.

- Physics and Variables Selection
- Individual study and study steps are listed in Table 19-2 and Table 19-3.

Using a Solution From Previous Study Steps

It is sometimes useful to run one physics simulation and then use the output of the first simulation as input into the second simulation. You can perform such a sequential study using multiple study steps in a single study or using multiple studies in the same model:

- Use two or more study steps in a sequence in the same study if you, for example, want to use a stationary or eigenvalue solution as the initial value for a time-dependent solution. You then add an **Eigenvalue** or **Stationary** study step node followed by a **Time Dependent** study step node. By default, COMSOL determines the values of the dependent variables in the fields for the physics heuristically depending on the model, so normally you do not have to take specific action. To specify the initial values of the dependent variables that you solve for, select the **Initial values of variables solved for** check box in the **Values of Dependent Variables** section. Then use the **Method** list to specify how to compute the initial values. Choose **Solution** and then select the solution to the stationary or eigenvalue problem solved using the previous study step. Doing so can be useful to specify a specific eigensolution to use as the initial value, for example.

  Another case is when you want to solve for two different physics, one at the time, but use the solution from the first study as input data for the physics in the second study. You then select the physics to solve for and the physics to disable in the solver under **Physics and Variables Selection** in the settings windows for each study step. Typically you solve for one physics (or set of physics) in the first study step and for the other physics in the other study step (you can also use two separate studies). To specify the values of dependent variables that you do not solve for, select the **Values of variables not solved for** check box in the **Values of Dependent Variable**s section. Then use the **Method** list to specify how to compute the values of variables not solved for. Choose **Solution** and then select the study step and solution to use.
- An example where you cannot just use one study is when a model modification is required in between the simulations. For example, if a boundary condition is changed, or if the initial value expressions needs to be changed. Another example when it is more practical to work with separate studies is when a study is inherently a multistep study, such as buckling in structural mechanics or modal analysis. Extending such studies with extra study steps normally becomes unnecessary complex. A third example is when doing optimization, where the problem without optimization often needs to be solved and tested separately from a study with optimization enabled.

See Values of Dependent Variables above for details about the settings for controlling the values of the dependent variables that you solve for or exclude from the solvers. Also see Physics and Variables Selection below for information about controlling which physics to solve for.

Physics and Variables Selection

All study step settings windows contain a **Physics and Variables Selection** section, which you can use to control which physics (or even specific variables and physics nodes) to solve for. This can be useful for:

- Solving physics in a sequence, including different physics features in each step.
- Solving and comparing different "analysis cases" for a model (sharing the same geometry and material) by varying boundary conditions, sources, or variables without the need to enable and disable nodes in the physics and recompute the solution.

By default, you can select from the participating physics. To select individual physics features and variables, select the **Modify physics tree and variables for study step** check box.

SELECTING PHYSICS TO SOLVE FOR

The **Physics** column contains the names of all physics in the model. You can choose to not solve for one or more of the physics by clicking the ✓ button in the **Solve for** column (by default, a study solves for all physics). The ✓ button then changes to an ✕ to indicate that the physics is not solved for, but it still provides values for the degrees of freedom (dependent variables) according to the settings for values of variables not solved for (see Values of Dependent Variables). Click the ✕ button to solve for the physics. In the **Discretization** list you can specify which discretization to use. The default (and often the only choice is **Physics settings**, which means that the study uses the discretization from the main physics node's settings. Changing it affects the discretization order used by this study. To add another discretization, use a separate **Discretization** node in the physics. The leftmost column is usually empty but contains a warning (⚠) if the physics' degrees of freedom are not solved for regardless of the setting in the **Solve for** column. This can be the case if the physics is not compatible with the study step.

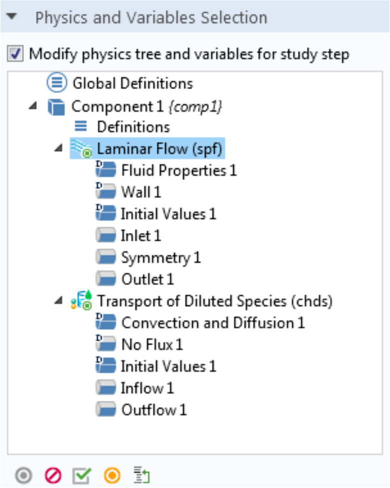
- The Add Physics Window
- The Add Study Window

SELECTING VARIABLES AND PHYSICS NODES TO INCLUDE

If the **Modify physics tree and variables for study step** check box is selected, you specify to select individual variables or physics features to include in the model that you solve. The **Physics and Variables Selection** section then contains a tree that is a copy of the following parts of the model tree in the Model Builder (see Figure 19-4):

- **Variables** nodes under **Global Definitions**.

- **Variables** nodes under **Component>Definitions** for all **Component** branches.
- All physics nodes in the **Component** branches.



▾ Physics and Variables Selection

☑ Modify physics tree and variables for study step

- Global Definitions
- Component 1 *{comp1}*
  - Definitions
  - Laminar Flow (spf)
    - Fluid Properties 1
    - Wall 1
    - Initial Values 1
    - Inlet 1
    - Symmetry 1
    - Outlet 1
  - Transport of Diluted Species (chds)
    - Convection and Diffusion 1
    - No Flux 1
    - Initial Values 1
    - Inflow 1
    - Outflow 1

Figure 19-4: An example of a Physics and Variables Selection section tree when the Modify physics tree and variables for study step check box is selected.

It is possible to include or exclude all variables, physics, and physics nodes in a study step (that are not disabled in the model tree). Select one or more nodes in the tree and right-click or use the buttons at the bottom of the section (below the tree) to change their status. Click the **Go to Source** button ( ) to move to the corresponding original node in the model tree. The following options are available:

Disabling and Enabling Physics and Variables Selection Nodes

Click **Disable** (or right-click to select from the context menu) to disable enabled nodes that are possible to disable. The contributions, conditions, or variables in a node that you disable are not included in the study when solving. You can also disable selected nodes by clicking the **Disable** button ( ) underneath the tree. A disabled node is unavailable in the tree.

Click **Enable** (or right-click to select from the context menu) to enable disabled nodes. The contributions, conditions, or variables in a node that you enable are included in the study when solving. You can also enable selected nodes by clicking the **Enable** button ( ) underneath the tree.

When you right-click, the following context menu options mean that a node cannot be enabled or disabled:

- **Cannot be Disabled**—for default nodes in the physics interfaces.
- **Disabled in Model Builder**—for nodes that you have disabled in the Model Builder.
- **Not Applicable**—for physics nodes that are not applicable for the study type in the study step. The item in the tree is not available.

When solving, equations and variables are generated as if the disabled nodes in the tree were disabled in the Model Builder. This means that the nodes' selections override each other as if the nodes were disabled in the Model Builder.

Change of States and Override and Contribution Indicators

An asterisk displays in the upper-right corner of nodes for which the state has been changed in the study step's selection tree compared to the state in the Model Builder. In this example, under the **Physics and Variables Selection** section, a Transport in Diluted Species interface ( ) is disabled (unavailable), provides no degrees of freedom (red dot in the lower-right corner), and has a change of state indicated by the asterisk. The asterisk means the Laminar Flow physics in the Model Builder is not disabled. Also see Figure 19-4 for another example. In general, any variable or physics node in the Model Builder that is disabled in any study step gets an asterisk in the upper-right corner. For physics this applies also when you have not selected the **Modify physics tree and variables for study step** check box and the physics is disabled in the **Solve for** column in the **Physics and Variables Selection** section.

The dynamic visual icon indicators for overridden and contributing nodes also appear in the tree in the settings window for the study steps when you have selected the **Modify physics tree and variables for study step** check box in a study step's settings window. When you select a physics node in the tree that appears, the override and contribution icon indicators appear in the same way as in the Model Builder when you select a physics node, but if you disable any physics node in the study step's tree, the icon indicators then show how the physics node override and contribute to the model when one of more physics nodes are disabled in the study step.

Options and States for the Physics

The following options are available for the main physics nodes under the **Physics and Variables Selection** tree. Right-click a node and select one of the following from the context menu or click the button beneath the tree (see Figure 19-4). Selecting these options affects the entire physics interface. Select:

- ☑ **Solve For** (the default setting) to solve for the physics, including all enabled physics nodes and the contributions, constraints, and variables that are added. This is similar to the ✓ button when you specify what physics to solve for without the selection tree.

  A physics interface in this state shows a small green circle in its lower-right corner to indicate that the study step solves for the degrees of freedom (dependent variables) in the physics. This is an example of a Laminar Flow physics with the green dot (≋).
- ◉ **Disable in Solvers** to not solve for the physics but provide degrees of freedom (dependent variables) and other physics variables using the settings for values of variables not solved for (see Values of Dependent Variables).

  A physics in this state shows a small yellow square in its lower-right corner to indicate that the study step provides degrees of freedom but does not solve for the physics. In this example, a Laminar Flow physics is both showing that it provides degrees of freedom (yellow dot in the lower-right corner) and has a change of state indicated by the asterisk (≋*).
- ⊘ **Disable in Model** to fully disable a physics in the model. The physics does not contribute to the study and no variables, including the degrees of freedom (dependent variables) are included.

  A disabled physics is unavailable and shows a small red square in its lower-right corner to indicate that the study step provides no degrees of freedom for it. In this example, a Transport of Diluted Species interface (*) is disabled (unavailable), provides no degrees of freedom (red dot in the lower right corner), and has a change of state indicated by the asterisk.

In addition, the physics can be in the following states:

- If the physics is disabled in the Model Builder, it is unavailable and shows a small red icon in its lower-right corner. If you right-click, the context menu contains **Disabled in Model Builder**. In this case, none of the options above is available.
- If the physics is not applicable because it does not support the study step, then by default it has the **Disable in Solvers** setting, and you can also choose **Disable in Model**. **Solve For** is not available.

Discretization Selection

You can also right-click a physics node in the selection tree to select the discretization. The discretizations appear at the bottom of the context menu (underneath the horizontal divider). In most case, the only option, and the default, is **Physics Settings**, which takes the discretization from the physics interface's settings window, but if you have added separate **Discretization** nodes, you can select from one of those instead of **Physics Settings**.

If you have the AC/DC Module, see Electric Shielding: model library path **ACDC_Module/Resistive_Devices/electric_shielding**.

Stationary

The **Stationary** ( ) study is used when field variables do not change over time.

For example, In electromagnetics, it is used to compute static electric or magnetic fields, as well as direct currents. In heat transfer, it is used to compute the temperature field at thermal equilibrium. In solid mechanics, it is used to compute deformations, stresses, and strains at static equilibrium. In fluid flow it is used to compute the steady flow and pressure fields. In chemical species transport, it is used to compute steady-state chemical composition in steady flows. In chemical reactions, it is used to compute the chemical composition at equilibrium of a reacting system.

It is also possible to compute several solutions, such as a number of load cases, or to track the nonlinear response to a slowly varying load.

A Stationary study node corresponds to a Stationary Solver (the default) or a parametric solver and is used to solve a stationary problem.

There is also an option to run a Stationary study with an Auxiliary sweep, with or without a continuation parameter. When a continuation parameter is selected the continuation algorithm is run, which assumes that the sought solution is continuous in these parameters. If no continuation parameter is given, a plain sweep is performed where a solution is sought for each value of the parameters. In both cases, a **Stationary Solver** node plus a **Parametric** attribute is used. The parametric solver is the algorithm that is run when a Parametric attribute node is active under a Stationary Solver. Similarly the adaptive solver is the algorithm that is run when an Adaptive Mesh Refinement node is active under a Stationary Solver.

Study Settings, **Results While Solving**, and **Mesh Selection** are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY EXTENSIONS

This section contains some optional extensions of the study, such as Auxiliary Sweep (including continuation), Adaptive Mesh Refinement, and load cases.

Load Cases

Select the **Define load cases** check box to define load cases as combinations of defined load groups, multiplied with optional weights (load factors), and constraint groups. When this check box is selected, and a Parametric attribute node is also used, the load cases are also displayed under the **Load Cases** section for the **Parametric** node.

Load cases are useful for efficiently solving for a number of cases with varying loads (and constraints) in the same model without the need to reassemble the stiffness matrix. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☰), and **Add** (+) buttons to make the list contain the load cases that you want to solve for. For each load case, click in the column for the load groups and constraint groups that you want to include in the load case. By default, no load groups and constraint groups are included (✕). Load groups and constraint groups that are included appear with a check mark (✓). Optionally, change the default weights for the load groups from 1.0 to another value in the corresponding **Weight** column (which is to the right of the load group that it is acting on). A weight of 1.5, for example, adds an extra 50% to the magnitude of the loads in the load group; a weight of –1 reverses the direction of the loads.

- Load Group and Constraint Group
- About the Parametric Solver
- Using Load Cases

Time Dependent

The **Time Dependent** ( ) study is used when field variables change over time.

For example, in electromagnetics, it is used to compute transient electromagnetic fields, including electromagnetic wave propagation in the time domain. In heat transfer, it is used to compute temperature changes over time. In solid mechanics, it is used to compute the time-varying deformation and motion of solids subject to transient loads. In acoustics, it is used to compute the time-varying propagation of pressure waves. In fluid flow, it is used to compute unsteady flow and pressure fields. In chemical species transport, it is used to compute chemical composition over time. In chemical reactions, it is used to compute the reaction kinetics and the chemical composition of a reacting system.

Selecting a Time Dependent study gives a solver with a Time-Dependent Solver. Use this study for a time dependent or transient simulation using a Time-Dependent Solver for computing the solution over time. Also see The Relationship Between Study Steps and Solver Configurations.

Mesh Selection, the **Include geometric nonlinearity** check box, Auxiliary Sweep, and Adaptive Mesh Refinement are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY SETTINGS

Select a **Time unit** from the list (default: s) to use a time unit that is convenient for the time span of the simulation. Then specify the time interval for the output from the simulation in the **Times** field using the selected time unit. You can type a monotonically increasing list of individual values, for example, `0 1 2 5 10 20`; use the `range` operator, for example, `range(0,0.1,1.5)`, which (using seconds as the time unit) gives time steps from 0 to 1.5 s with a step size of 0.1 s; or use any combinations of such input.

Select the **Relative tolerance** check box to override the relative tolerance suggested by the program. The tolerance settings control the internal time steps taken by the solver, so selecting large time steps for the output times does not affect the accuracy in the time stepping.

When plotting the results from a time-dependent simulation you can choose to plot the solution at any of the times specified in the **Times** field. You can also plot an interpolated solution at any intermediate time. The interpolation used between times is a cubic Hermite spline; that is, the interpolation uses both the solution values and their time derivatives at two points: the closest output times before and after the time for which the interpolated solution is computed.

RESULTS WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** and **Update at** lists. The software plots the data set of the selected plot group as soon as the results become available either at the times specified by the output times (from the **Times** field) or at a set of internal times defined by the solver. Select **Output times** (the default) or **Time steps from solver** from the **Update at** list.

Use the Probes list to select any probes to evaluate. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✕), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving. For the probes you also select **Output times** or **Time steps from solver** (the default) from the associated **Update at** list at the bottom of the section.

STUDY EXTENSIONS

This section contains some optional extensions of the study. The options are mutually exclusive and only one of the check boxes can be selected. See Auxiliary Sweep and Adaptive Mesh Refinement for those common settings.

Automatic Remeshing

Select the **Automatic remeshing** check box if you want the solver to remesh automatically when the quality of the mesh becomes poor in a Time Dependent study. Select the geometry to use for the automatic remeshing from the **Remesh in geometry** list With automatic remeshing active, the solver adds an **Automatic Remeshing** subnode under the **Time-Dependent Solver** node. In that subnode you specify the mesh quality expression that determines when to remesh.

Time Discrete

Selecting the **Time Discrete** ( ) study gives a study step with a Time Discrete Solver. Use it for performing time-dependent analysis using the projection method. The settings for this study node are the same as for the Time Dependent node. Also see The Relationship Between Study Steps and Solver Configurations.

Time-Dependent Modal

The **Time-Dependent Modal** ( ) study is used to compute the dynamic structural deformation of an object subject to a transient force.

The study consists of two study steps: one study step for computing the eigenfrequencies and eigenmodes of the structure, and a second study step for computing the modal response. In the modal superposition analysis, the deformation of the structure is represented by a linear combination of the structure's eigenmodes. This means that the frequency content of the loads is limited by the frequencies of the computed eigenmodes. All loads are assumed to have the same variation with time. A Time-Dependent Modal study usually results in a faster computation than a direct solution using the Time Dependent study.

Selecting a Time-Dependent Modal study gives a study step with a Modal Solver. Use it for performing transient response analyses. The settings are the same as for the Time Dependent node for the sections that they share (the Time Dependent study includes some additional settings). Also see The Relationship Between Study Steps and Solver Configurations.

With the Structural Mechanics Module, see Various Analyses of an Elbow Bracket, model library path **Structural_Mechanics_Module/Tutorial_Models/elbow_bracket**.

Eigenfrequency

The **Eigenfrequency** ( ) study is used to compute eigenmodes and eigenfrequencies of a linear or linearized model.

For example, in electromagnetics, the eigenfrequencies correspond to the resonant frequencies and the eigenmodes correspond to the normalized electromagnetic field at the eigenfrequencies. In solid mechanics, the eigenfrequencies correspond to the natural frequencies of vibrations and the eigenmodes correspond to the normalized deformed shapes at the eigenfrequencies. In acoustics, the eigenfrequencies correspond to the resonant frequencies and the eigenmodes correspond to the normalized acoustic field at the eigenfrequencies.

Selecting an Eigenfrequency study gives a solver with an Eigenvalue Solver. Use this study to solve an eigenvalue problem for a set of eigenmodes and associated eigenfrequencies. Also see The Relationship Between Study Steps and Solver Configurations.

Mesh Selection, the **Include geometric nonlinearity** check box, and **Study Extensions** (Adaptive Mesh Refinement) are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY SETTINGS

Use the **Desired number of eigenfrequencies** field to specify the number of eigenfrequencies you want the solver to return.

In the **Search for eigenfrequencies around** field, you can specify a value around which the eigenvalue solver should look for solutions to the eigenvalue equation.

Use the **Eigenfrequency search method around shift** list to control how the eigenvalue solver searches for eigenfrequencies around the specified shift value:

- Select **Closest in absolute value** (the default value) to search for eigenfrequencies that are closest to the shift value when measuring the distance as an absolute value.
- Select **Larger real part** to search for eigenfrequencies with a larger real part than the shift value.
- Select **Smaller real part** to search for eigenfrequencies with a smaller real part than the shift value.
- Select **Larger imaginary part** to search for eigenfrequencies with a larger imaginary part than the shift value.
- Select **Smaller imaginary part** to search for eigenfrequencies with a smaller imaginary part than the shift value.

The Eigenvalue Solver can in some cases return more than the desired number of eigenfrequencies (up to twice the desired number). These are eigenfrequencies that the eigenvalue solver finds without additional computational effort.

Tuning Fork: model library path **COMSOL_Multiphysics/Structural_Mechanics/tuning_fork**.

Eigenvalue

The **Eigenvalue** ( ) study is used to compute the eigenvalues and eigenmodes of a linear or linearized model in a generic eigenvalue formulation where the eigenvalues are not necessarily frequencies. The Eigenvalue study gives you full control of the eigenvalue formulation, in contrast to the eigenfrequency study that is adapted for specific physics interfaces. The Eigenvalue study is typically used for equation-based modeling.

Selecting an Eigenvalue study gives a solver configuration with an Eigenvalue Solver.

Mesh Selection, the **Include geometric nonlinearity** check box, and **Study Extensions** (Adaptive Mesh Refinement) are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY SETTINGS

Use the **Desired number of eigenvalues** field to specify the number of eigenvalues you want the solver to return.

In the **Search for eigenvalues around** field, you can specify a value around which the eigenvalue solver should look for solutions to the eigenvalue equation.

Use the **Eigenvalue search method around shift** list to control how the eigenvalue solver searches for eigenvalues around the specified shift value:

- Select **Closest in absolute value** (the default value) to search for eigenvalues that are closest to the shift value when measuring the distance as an absolute value.
- Select **Larger real part** to search for eigenvalues with a larger real part than the shift value.
- Select **Smaller real part** to search for eigenvalues with a smaller real part than the shift value.
- Select **Larger imaginary part** to search for eigenvalues with a larger imaginary part than the shift value.
- Select **Smaller imaginary part** to search for eigenvalues with a smaller imaginary part than the shift value.

The Eigenvalue Solver can in some cases return more than the desired number of eigenvalues (up to twice the desired number). These are eigenvalues that the eigenvalue solver finds without additional computational effort.

Frequency Domain

The **Frequency Domain** ( ) study is used to compute the response of a linear or linearized model subjected to harmonic excitation for one or several frequencies.

For example, in solid mechanics, it is used to compute the frequency response of a mechanical structure with respect to particular load distributions and frequencies. In acoustics and electromagnetics, it is used to compute the transmission and reflection versus frequency. A Frequency Domain study accounts for the effects of all eigenmodes that are properly resolved by the mesh and how they couple with the applied loads or excitations. The output of a Frequency Domain study is typically displayed as a transfer function, for example, magnitude or phase of deformation, sound pressure, impedance, or scattering parameters versus frequency.

It is also possible to add an auxiliary sweep to this study, which creates a multiparameter sweep (Parametric solver) over both the frequency and the given parameters, and optionally with continuation in the frequency or in one of the given parameters. It corresponds to a stationary parametric solver that is preset to linearize the equations (Stationary Solver with a Parametric attribute).

Alternatively, select the **Use asymptotic waveform evaluation** check box to use an AWE Solver instead of the Parametric solver.

> The **Include geometric nonlinearity** check box, **Results While Solving**, **Mesh Selection**, and Auxiliary Sweep are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY SETTINGS

Specify the frequencies to use for the frequency sweep. Type the frequencies in the **Frequencies** field using space-separated numbers or the `range` function.

Use the **Load parameter values** field to select a file with parameter values. You can browse your file system for files by clicking the **Browse** button. After selecting a file click the **Read File** button to load the parameter values into the **Frequencies** field.

STUDY EXTENSIONS

Also see Auxiliary Sweep.

Asymptotic Waveform Evaluation

Select the **Use asymptotic waveform evaluation** check box to enable the *asymptotic waveform evaluation (AWE) solver*. The Frequency Domain study generates a solver configuration that is used to solve a stationary parametric problem or an asymptotic waveform evaluation problem. By selecting this check box, this study step corresponds to an AWE Solver.

Frequency Domain Modal

The **Frequency Domain Modal** ( ) study is used to compute the response of a linear or linearized structural mechanics model subjected to harmonic excitation for one or several frequencies.

The study consists of two study steps: an Eigenfrequency study step for computing the eigenfrequencies and eigenmodes of the structure followed by a second Frequency-Domain Modal study step for computing the modal response. In the modal superposition analysis, the deformation of the structure is represented by a linear combination of the structure's eigenmodes. This means that the frequencies to be studied are limited by the frequencies of the computed eigenmodes. A Frequency Domain Modal study usually results in a faster computation than a direct solution using the Frequency Domain study.

The Frequency Domain Modal study node corresponds to a modal frequency sweep for systems with frequency-based loads. It gives a Modal Solver.

> The **Include geometric nonlinearity** check box and **Mesh Selection** are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

STUDY SETTINGS

Specify the frequencies to use for the frequency sweep. Enter the frequencies in the **Frequencies** field using space-separated numbers or the `range` function.

Use the **Load parameter values** field to select a file with parameter values. Click the **Browse** button to browse the file system. After selecting a file click the **Read File** button to load the parameter values into the **Frequencies** field.

With the Structural Mechanics Module, see Various Analyses of an Elbow Bracket, model library path **Structural_Mechanics_Module/Tutorial_Models/elbow_bracket**.

Batch

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

Use a **Batch** ( ) study to start a COMSOL Multiphysics batch process that solves the current study on your computer. Once the filename and directory are set, right-click the parent **Study** node and choose **Compute** = to start a COMSOL batch process that computes the current study.

You cannot create another Batch, Batch Sweep, Cluster Sweep, or Cluster Computing in the same study at the same time.

When the batch process starts the COMSOL Desktop follows the progress in The External Process Window. When the process finishes (or you click the **Detach Job** button to stop the process) an **External Process** node is added, one for each parameter, under a **Batch Data** node as in Figure 19-5. The External Process node represents the current running process. For more information, see Batch (Job Configuration), Batch Data, and External Process.

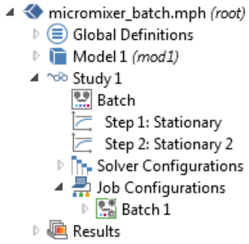


Figure 19-5: An example of the two kinds of Batch nodes available under Study and Job Configurations.

Micromixer—Batch Version: model library path: **COMSOL_Multiphysics/Tutorial_Models/micromixer_batch**

BATCH SETTINGS

Specify the file where to store the model in the **Filename** field. Choose the directory where to store the model in the **Directory** field by typing it directly or clicking the **Browse** button to choose a batch directory. If you are connected to a COMSOL server on another computer, you can control the working directory used by the COMSOL server if you select the **Specify COMSOL server directory path** check box and enter the path to the server **Directory** or **Browse** for the path. Otherwise a temporary directory on the COMSOL server is used to save files.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that create graphical contents such as exporting to file. Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before is must inform that it is still running. Failure to do so means that the process is considered dead and a new process is started if not the maximum number of job restarts is reached.

Batch Sweep

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

Use the **Batch Sweep** ( ) study to find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest. If you want to make a full multiparameter sweep (solving first for the first value of the first parameter combined with all values of the second parameter, then for the second value of the first parameter combined with all values of the second parameter, and so on), you can add several Parametric Sweep nodes, one for each additional parameter except the outermost parameter. COMSOL then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** node names.

The Batch Sweep is always the outermost sweep. It starts multiple COMSOL Multiphysics batch processes—one for each parameter set in the Batch Sweep node—that solves the current study on your computer given the parameter set.

Once the filename and directory are set, right-click the parent **Study** node and choose **Compute** = to start the COMSOL batch processes that computes the current study.

When the batch sweep process starts the COMSOL Desktop follows the progress in The External Process Window. When the process finishes (or you click the **Detach Job** button to stop the process) an External Process node, one for each parameter, is added under a Batch Data node as in Figure 19-5. The External Process node represents the current running process.

If you click **Detach Job**, the **Batch Sweep** then no longer synchronizes the solutions and accumulated probe table. To reenable the synchronization use the **Show All Progress** button in the **Batch Data** node.

!

You cannot create another Batch, Batch Sweep, Cluster Sweep, or Cluster Computing in the same study at the same time.

STUDY SETTINGS

The batch sweep is a multiparameter sweep with its parameters solved as a batch job; see Parametric Sweep for more information.

OUTPUT WHILE SOLVING

Use the Probes list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** ( ), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so on) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not selected, the probes selected by the **Probes** selector are used.

> No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to make so-called response surfaces directly from the **Results** view toolbar **Surface Plot** button.

BATCH SETTINGS

Specify the file where to store the model in the **Filename** field. Choose the directory where to store the model in the **Directory** field by typing it directly or clicking the **Browse** button to choose a batch directory.

Select the **Clear meshes** check box to clear the meshes before running the batch sweep. The default is to clear the meshes. Select the **Clear solutions** check box to clear the solutions before running the batch sweep. The default is to clear the meshes. Select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional postprocessing to be performed after the sweep has finished. The setting is similar to the **All** and **Last** setting in the **Memory settings for jobs** for Parametric Sweep. The default is to disable solution synchronization. Select the **Synchronize accumulated probe table** check box to synchronize the accumulated probes computed by the batch processes with the model. The accumulated probe synchronization is enabled by default. Select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases the solution synchronization and probe synchronization functionality should be used instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that create graphical contents such as exporting to file. Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before is must inform that it is still running. Failure to do so means that the process is considered dead, and a new process is started if the maximum number of job restarts is not reached.

Cluster Computing

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

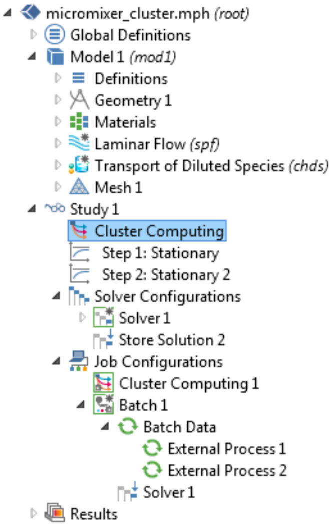
Use the **Cluster Computing** ( ) study when you want to submit COMSOL Multiphysics batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster. For more information, see Cluster Computing (Job Configuration) and its related functionality. Also see Figure 19-6.

Use Cluster Computing to utilize a cluster or cloud to solve a single large model using distributed memory. For maximum performance, the COMSOL cluster implementation can utilize shared-memory multicore processing on each node in combination with the Message Passing Interface (MPI) based distributed memory model. This brings a major performance boost by making the most out of the computational power available.

It is also intended as an interface for setting up distributed COMSOL batch jobs on the computer where the COMSOL Desktop is running. If you are running in distributed mode interactively, it is usually not needed unless you want a simple way to distribute parametric sweep.

Once you have specified the settings, right-click the main **Study** node and select **Compute** = in order to start a COMSOL process that solves the current study.

When the cluster computing process starts the COMSOL Desktop follows the progress in The External Process Window. When the process finishes (or you click the **Detach Job** button to stop the process) an External Process node is selected that represents the current running process. If you are running COMSOL in distributed mode the model runs in the current process. In this case only the **Distribute parametric sweep** check box is selected.



- micromixer_cluster.mph *(root)*
  - Global Definitions
  - Model 1 *(mod1)*
    - Definitions
    - Geometry 1
    - Materials
    - Laminar Flow *(spf)*
    - Transport of Diluted Species *(chds)*
    - Mesh 1
  - Study 1
    - Cluster Computing
    - Step 1: Stationary
    - Step 2: Stationary 2
    - Solver Configurations
      - Solver 1
      - Store Solution 2
    - Job Configurations
      - Cluster Computing 1
      - Batch 1
        - Batch Data
          - External Process 1
          - External Process 2
        - Solver 1
  - Results

Figure 19-6: An example of the two kinds of Cluster Computing nodes available under Study and Job Configurations.

Micromixer—Cluster Version: model library path:
COMSOL_Multiphysics/Tutorial_Models/micromixer_cluster

You cannot create another Batch, Batch Sweep, Cluster Sweep, or Cluster Computing in the same study at the same time.

BATCH SETTINGS

After making these settings, click the **Save as Default** (💾) button on the settings window toolbar to save the current directory settings as the default preference.

Choose the **Cluster type**—**General** (the default), **HPCS 2008**, **WCCS 2003**, **OGS/GE**, or **Not distributed**:

General

Select **General** (the default) to configure to run on many types of clusters and schedulers, including Linux clusters.

- When **General** is selected, and you have started a multiprocessor daemon (MPD) on the computer, click to select the **MPD is running** check box.
- The entry in the **Host file** field specifies the host file used for the job. If left empty, MPD looks for a file `mpd.hosts` in the Linux home directory.
- Select which bootstrap server MPI should use in the **Bootstrap server** setting.

- If your cluster is Linux and it requires that an SSH (secure shell) or an RSH (remote shell) is installed in an uncommon directory, use the **Rsh** field to set the RSH communication protocol.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter details for the Filename and Directory Settings for all Cluster Types and Use Batch License Settings for all Cluster Types.

HPCS 2008 or WCCS 2003

Select **HPCS 2008** to use the Windows HPC Server 2008 job scheduler to submit the batch job. Select **WCCS 2003** to use the Windows Compute Cluster Server 2003 job scheduler to submit the batch job.

When **WCCS 2003** or **HPCS 2008** is selected:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter details for the Filename and Directory Settings for all Cluster Types and Use Batch License Settings for all Cluster Types.

OGS/GE

Select **OGS/GE** to use the open grid scheduler/grid engine job scheduler to submit the batch job. When **OGS/GE** is selected:

- Select which bootstrap server should be used by MPI using the **Bootstrap server** setting.
- If your cluster is Linux and it requires that an SSH (secure shell) or an RSH (remote shell) is installed in an uncommon directory, use the **Rsh** field to set the RSH communication protocol.
- Specify the **Number of slots** and the name of the scheduler queue in **Queue name**.
- Enter the **Queue name** for the cluster queue.
- Enter details for the Filename and Directory Settings for all Cluster Types and Use Batch License Settings for all Cluster Types.

Not Distributed

Select **Not distributed** when you want to submit a batch job to a job scheduler without running a distributed job. Enter details for the Filename and Directory Settings for all Cluster Types and Use Batch License Settings for all Cluster Types.

Filename and Directory Settings for all Cluster Types

Specify the file where to store the model in the **Filename** field. Choose the mounted file system directory where to store the model in the **Directory** field by entering it directly or by clicking the **Browse** button to choose a batch directory.

- If you are connected to a COMSOL server on another computer, you can control the working directory used by the COMSOL server if you select the **Specify COMSOL server directory path** check box and enter the path to the server **Directory** or **Browse** for the path. Otherwise a temporary directory on the COMSOL server is used to save files.
- If the batch job has another path to the directory, select the **Specify external COMSOL batch directory path** check box and enter the path to the external process (batch) directory in the **Directory** field or click **Browse**. The cluster job uses this path from the compute node to access the input file and write back the result. On Windows this must be a fully qualified UNC path, for example, `\\head1\shared\clusterprojects`.

- If COMSOL is installed in another directory than where the COMSOL Desktop runs, select the **Specify external COMSOL installation directory path** check box and then specify the installation directory (click **Browse** or enter the path to the **Directory**). This can occur if you are submitting jobs to a job scheduler. Typically on Windows this is the UNC path to the COMSOL installation root directory for the compute nodes to access the required COMSOL binaries—for example, `\\head1\shared\COMSOL44`.

Use Batch License Settings for all Cluster Types

Select the **Use batch license** check box to run using batch licenses. Batch licenses can be used to run multiple batch jobs for different models that only depend on a parameter. Usually you should use the **Cluster Sweep** node.

REMOTE AND CLOUD ACCESS

Enable the **Run remote** check box if you want to run COMSOL on a remote machine using a remote start command such as SSH and using a file transfer program such as SCP to transfer the files to and from the remote computer. This allows you to run on a machine installed on your network without a client server connection or on a machine installed on a remote cloud.

COMSOL must be installed on the remote machine and all settings must be specified correctly in the **Specify external COMSOL batch directory path** and **Specify external COMSOL installation directory path** settings according to how COMSOL is installed on the remote machine and the remote machines working directories.

You must also be able to access the remote machine without a password using the access method selected. This can be achieved by using something that does not require a password, for example, SSH or similar.

Choose the method for starting COMSOL remotely in **Remote invoke command**. You can choose **None**, **SSH**, or **User** defined. For the **SSH** setting you can choose between using **SSH**, **Putty**, or a **User** defined SSH command. You can select the SSH commands installation directory in the SSH directory if the SSH command is not available on the PATH. The SSH key file directory is set in the **SSH key file** setting. You can set ports you want to forward in **Forward ports** and the host you want to forward the ports to in **Port host**. This is useful when you have the license manager installed locally, but the machine where COMSOL is running cannot access the license manager; for instance, if the machine is in the cloud. Specify the user name to use on the remote machine with SSH user. For the **User** defined **Remote invoke command** you can enter a command in the **Command** setting. In the **Command** setting any use of the keyword *`{remotehost}`* is replaced by the name of the remote host when COMSOL starts.

Choose the method for transferring files to the remote computer in **File transfer command**. You can choose **None**, **SCP**, or **User** defined. For the **SCP** setting you can choose between using **SCP**, **Putty**, or a **User** defined SCP command. You can select the SCP commands installation directory in **SCP directory** if the SCP command is not available on the PATH. The SCP key file directory is set in the **SCP key file** setting. Specify the user name to use on the remote machine with SCP user. For the User defined **File transfer command** you can enter a command in the **To remote command** for the transfer of files from the local computer to the remote machine and **From remote command** for the transfer of files from the remote machine to the local computer. In the settings any use of the keyword *`{remotehost}`* is replaced by the name of the remote host. Any use of the keyword *`{localfile}`* is replaced by the name of the local file, and any use of the keyword *`{remotefile}`* is replaced by the name of the remote file.

Use the **Remote hosts** to list the host you want to run on. If several hosts are listed COMSOL allocates a job on the first host that is free. Use the **Remote OS** to specify if the remote computer is running the same OS (**Native**) or is running **Linux** or **Windows**.

For information about using COMSOL with the Amazon Elastic Compute Cloud™ for cloud-based computing, see the separate document *Running COMSOL on the Amazon Cloud*, which is available as a PDF file in the online documentation and on the Windows® start menu.

CLUSTER SETTINGS

If you are running a parametric sweep and want to distribute it by sending one parameter value to each compute node, select the **Distribute parametric sweep** check box. This requires that your study includes a parametric sweep.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that creates graphical contents such as exporting to file.

Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously.

Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete.

Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before is must inform that it is still running. Failure to do so means that the process is considered dead, and a new process is started unless the maximum number of job restarts is reached.

You can also make changes to how some of these settings are displayed throughout COMSOL in The Preferences Dialog Box in the **Remote Computing** and **Multicore and Cluster Computing** sections.

Running COMSOL as a Client/Server

Cluster Sweep

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

Use the **Cluster Sweep** ( ) study to solve several models in parallel where each model has a different set of Parameters. For example, find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest. The cluster sweep can include multiple independent parameters directly for a full multiparameter sweep (solve for the first value of the first parameter combined with all values of the second parameter, then the second value of the first parameter combined with all values of the second parameter, and so on, or use a specified combination of parameter values).

You can also add more than one Parametric Sweep node to create nested parametric sweeps. COMSOL then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** node names. The Cluster Sweep is always the outermost sweep. It starts multiple COMSOL batch processes, one for each parameter set in the **Cluster Sweep** node, that solves the current study on your cluster given the parameter set by submitting COMSOL batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster.

For more information, see Cluster Computing (Job Configuration) and its related functionality. If you are running in distributed mode interactively, it is usually not needed.

You cannot create another Batch, Batch Sweep, Cluster Sweep, or Cluster Computing in the same study at the same time.

Right-click the **Study** node to add a **Cluster Sweep** study node to run a batch jobs on a cluster. Once you have specified the settings, right-click the main **Study** node and select **Compute** = in order to start a COMSOL batch process that solves the current study for the given parameter sets.

When the cluster sweep process starts the COMSOL Desktop follows the progress in The External Process Window. When the process finishes (or you click the **Detach Job** button to stop the process) an External Process node, one for each parameter, is added under a Batch Data node as in Figure 19-5. The External Process node represents the current running process.

Click the **Save as Default** button (💾) in the settings window toolbar to save the current setting as default. If you are running COMSOL in distributed mode the model runs in the current process.

STUDY SETTINGS

The cluster sweep is a multiparameter sweep with its parameters solved as a distributed batch job; see Parametric Sweep for more information about parameter sweeps.

OUTPUT WHILE SOLVING

This section is not available if the study also contains a Parametric Sweep.

Use the **Probes** list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☒), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.

No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to make response surfaces directly from the **Results** view toolbar **Surface Plot** button.

BATCH SETTINGS

Except for the information below, see Batch Settings (for the Cluster Computing node) for settings details.

Before Sweep

Under **Before sweep**, the **Clear meshes** check box is selected by default to clear the meshes before running the batch sweep. The **Clear solutions** check box is selected by default to clear the solutions before running the batch sweep. Click to clear one or both of the check boxes as required.

During Sweep

Under **During sweep**, click to select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional analysis to be performed after the sweep has finished.

The setting is similar to the **All** and **Last** setting in the **Memory settings for jobs** for a **Parametric Sweep**.

The **Synchronize accumulated probe table** check box is selected by default to synchronize the accumulated probes computed by the batch processes with the model. Click to clear the check box if required.

After Sweep

Under **After sweep**, select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases the solution synchronization and probe synchronization functionality should be used instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently.

REMOTE AND CLOUD ACCESS

See Remote and Cloud Access described for Cluster Computing.

STUDY EXTENSIONS

See Study Extensions described for Cluster Computing.

Modal Reduced Order Model

The **Modal Reduced Order Model** ( ) study is used for exporting reduced order model matrices.

The study consists of two study steps: an Eigenfrequency study step, for computing the eigenfrequencies and eigenmodes of a linear or linearized model in the frequency domain, followed by a **Modal Reduced Order Model** study step, for exporting the reduced order model matrices.

A Modal Reduced Order Model study (and study step) uses the Modal Solver. It uses the transient equation form for the export. The matrices can be accessed through the COMSOL API or in table format from a System Matrix derived value.

The **Include geometric nonlinearity** check box and **Mesh Selection** are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

Multigrid Level

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**. Then right-click any of the **Study Step** nodes to add a **Multigrid Level** ( ) node, which specifies the geometric multigrid level used by the study.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics by clearing the **Use in this study** check box (selected by default). In the **Discretization** list, you can specify which discretization to use. Changing it affects the discretization order used by this study. You have to add the discretization you want to use in the physics.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

Parametric Sweep

Use a **Parametric Sweep** ( ) study to find the solution to a sequence of stationary or time-dependent problems that arise when you vary some parameters of interest. The parametric sweep can include multiple independent parameters directly for a full multiparameter sweep (solve for the first value of the first parameter combined with all values of the second parameter, then the second value of the first parameter combined with all values of the second parameter, and so on, or use a specified combination of parameter values). You can also add more than one **Parametric Sweep** node to create nested parametric sweeps. The program then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** nodes’ names.

> **!** It is only possible to use one Sensitivity, Optimization, or **Parametric Sweep** feature in any study.

The settings window has the following sections.

STUDY SETTINGS

Use the **Sweep type** list to specify the type of sweep to perform. The **Specified combinations** type (the default) solves for a number of given combination of values, while the **All combinations** type solves for all combination of values. Using all combinations can lead to a very large number of solutions.

Use the **Parameter names** and **Parameter value list** table to specify parameter names and values for the parametric solver. Use the **Add** button (+) to add a row to the table. Each row has one parameter name and a corresponding parameter value list. For the **Specified combinations** sweep type, the list of values must have equal length. When you click in the **Parameter value list** column to define the parameter values you can click the **Range** button ( ) to define a range of parameter values.

If more than one parameter name have been specified the lists of parameter values are interpreted as follows: Assume that the parameter names are `p1` and `p2`, and that `p1` has the list 1 3 and `p2` has the list 2 4. For the **Specified combinations** sweep type, the solver first uses `p1` equal to `1` and `p2` equal to `2`. Thereafter, it uses `p1` equal to `3` and `p2` equal to `4`. And when the sweep type in **All combinations**, the solver uses the following order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. Use the **Load from File** button ( ) to browse to such a text file. The program appends the read names and values to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values, and a space separating the values. Click the **Save to File** button (💾) to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®).

OUTPUT WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** list. The software plots the data set of the selected plot group as soon as the results become available.

Use the **Probes** list to select probes to update during the parametric sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☰), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for parametric sweep. Note that the control of tables and plot windows are controlled by the probe settings.

> ! If a probe is updated on the **Parametric Sweep** level and also through another solution process (for example, a time-dependent simulation) this probe is not updated at the **Parametric Sweep** level. When the probes themselves (not the probe expression) depend on model parameters, the update of these probes is only correct for parameter sweeps that are done through outer parametric sweeps (not by a parametric solver). Outer parametric sweeps are performed by a **Parametric** node under **Job Configurations**. COMSOL Multiphysics currently does not autodetect model parameters in probes, so you might want to select **Off** from the **Use parametric solver** list in the **Study Extensions** section for the **Parametric Sweep** study node.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the parametric sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.

> No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to modify the table data and make so-called response surfaces directly from the **Results** view toolbar **Surface Plot** button.

You can use the **Memory settings for jobs** and the **Keep solutions in memory** list to control how to store the solutions from the individual parametric sweep solutions. Select **All** to store all the parametric sweep solutions in memory, or select **Only last** to store only the last solution from the parametric sweep. When only the last solution is stored you can also select the **Save each solution as model file** check box. It stores the separate parametric sweep solutions and their corresponding models in separate MPH-files. Enter a filename in the **Filename** field or click **Browse** to choose a name and location for the model files. You can also use probes to collect some solution values of interest during the sweep rather than storing all solutions, which can save memory and solution time.

STUDY EXTENSIONS

From the **User parametric solver** list, select one of the following options:

- **Automatic** (the default) to generate a **Parametric** job configuration, unless the problem and parameters are such that the parametric sweep can be realized through a **Stationary Solver** with a **Parametric** solver node ( ), which is more efficient.
- **Off** to always generate a **Parametric** job configuration.

To reduce the size of MPH-files for models using parametric sweeps, you have the option of storing only the last solution in the sweep in the file. If you want to use this setting as the default, open The Preferences Dialog Box and click **Parametric Sweep**. Then choose **Only last** from the **Keep solutions in memory** list. You then have the further option of saving each solution as a model file. To do this, select the **Save each solution as model file** check box and then enter a filename in the **Filename** field, or click **Browse** to choose a name and location for the model files. The default option in the **Keep solutions in memory** list, **All**, stores all solutions in the file.

- Probes
- Stationary for information about the continuation parametric solver versus parametric sweeps.
- Job Configurations
- Using a Job Configuration to Store Parametric Results on File

Sensitivity

To enable this option in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

Use the **Sensitivity** ( ) study step to add a sensitivity analysis to the study. Using a Sensitivity study node you can add sensitivity functions at the study level and use model parameters as global control variables. Compared to The Sensitivity Interface, the Sensitivity study step has these capabilities:

- Model parameters can act as global control variables
- Sensitivity functions can be added at the study level
- Sensitivity functions and variables from physics interfaces can be used on the study level.

It is only possible to use one of the Sensitivity, Optimization, or Parametric Sweep study steps in any study.

Sensitivity Analysis of a Communication Mast Detail: model library path **COMSOL_Multiphysics/Structural_Mechanics/mast_diagonal_mounting_sensitivity**.

The settings window has the following sections:

Sensitivity Method

Choose a method from the **Gradient method** list: one of the analytical methods **Forward** or **Adjoint** (the default). See Choosing a Sensitivity Method for information about the forward and adjoint methods. These methods have similar limitations as the gradient-based optimization methods (SNOPT and Levenberg-Marquardt). For example, nonanalytic functions are not correctly treated. Also, when model parameters are used in the geometry or mesh, the sensitivity is not correctly computed.

You choose the study step to use from the **Study step** list, which contains **None** and any supported study steps in the study. The supported study step types are Stationary, Time Dependent, and Frequency Domain.

Objective Function

You specify the objective function for the optimization problem in the table's **Expression** column. Enter any globally available expression that evaluates to a real number. Optionally, you can add a description in the **Description** column. Click the **Add Expression** (+) and **Replace Expression** ( ) buttons to search through a list of predefined expressions.

For a sensitivity objective that is expressed in terms of the solution u of a PDE (or in terms of control variables), Integration is one example of how you can define a scalar objective as required by the sensitivity solver. The evaluation of the objective function is similar to Global Variable Probe, so any variable that can be represented by a global variable probe is suited as an objective.

Multiple Objectives

If you have defined more than one objective function, choose how to evaluate the overall objective: For sensitivity studies, only **Sum of objectives** is available.

Solution

Here you select the evaluation method of the objective function when several solutions are present, like for Time Dependent studies. For sensitivity studies, only **Auto** is available. The solver chooses the evaluation method based on the innermost study. For studies in the Frequency Domain, the contributions from all solutions are summed (equivalent to the **Sum of objectives** option). For a Time Dependent study, the optimization solver selects the last solution (final time).

Control Variables and Parameters

The table under **Control Variables and Parameters** is used to define control variables. In this table you can select all parameters defined in the **Global Definitions>Parameters** node's settings window through the **Add** (+) button.

From a list in the **Parameter name** column, select the parameter to define as a control variable.

Move control parameter rows up and down using the **Move Up** (↑) and **Move Down** (↓) buttons. To remove a control parameter, select some part of that parameter's row in the table and click the **Delete** button ( ). You can also save the definitions of the control parameters to a text file by clicking the **Save to File** button ( ) and using the **Save to File** dialog box that appears. To load a text file with control variables, use the **Load from File** button ( ) and using the **Load from File** dialog box that appears. Data must be separated by spaces or tabs.

The tables to activate or deactivate objective functions and control variables in the model are only visible if those functions or variables are present in the model.

If you have the LiveLink™ *for* Excel®, you can also save and load control variables to and from Microsoft Excel Workbook (*.xlsx) files.

Boundary Mode Analysis

The **Boundary Mode Analysis** ( ) study is used to compute the propagation constants or wave numbers as well as propagating mode shapes, for a given frequency at a port.

The study combines a Boundary Mode Analysis study step at a port (boundary) (which can represent, for example, a cross section of a waveguide) with a Frequency Domain study step for the full geometry.

This study is available with the Electromagnetic Waves or Microwave Heating interfaces, which both require either the RF Module or the Wave Optics Module.

STUDY SETTINGS

- Enter a **Desired number of modes**. The default is 1.
- Select a method to **Transform**—**Effective mode index** (the default), **Out-of-plane wave number**, or **None**.
- Enter a value or expression in the **Search for modes around** field. The default is 1.
- Enter a **Port name**. The default is 1.
- Enter a value or expression for the **Mode analysis frequency**. The default is 1GHz.
- Use the **Mode search method around shift** list to control how the solver searches for modes around the specified value to search around:
  - Select **Closest in absolute value** (the default value) to search for modes that are closest to the search value when measuring the distance as an absolute value.
  - Select **Larger real part** to search for modes with a larger real part than the search value.
  - Select **Smaller real part** to search for modes with a smaller real part than the search value.
  - Select **Larger imaginary part** to search for modes with a larger imaginary part than the search value.
  - Select **Smaller imaginary part** to search for modes with a smaller imaginary part than the search value.

The **Include geometric nonlinearity** check box, **Mesh Selection**, and **Study Extensions** are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

- With the RF Module, see Polarized Circular Ports, model library path **RF_Module/Tutorial_Models/polarized_circular_ports**.
- With the Wave Optics Module, see Dielectric Slab Waveguide, model library path **Wave_Optics_Module/Verification_Models/dielectric_slab_waveguide**.

Coil Current Calculation

The **Coil Current Calculation** ( ) study is used to compute the current of a Multi-Turn Coil Domain and produces a current density corresponding to a bundle of conductive wires.

The **Coil Current Calculation** study and study step is available for 3D models using the Magnetic Fields interface and the Multi-Turn Coil Domain node (which requires the AC/DC Module). Use it to solve an eigenvalue problem for the current flow in a Multi-Turn Coil Domain node that gives the current density likely produced by a bundle of conductive wires. The best results are obtained when the coil has a constant cross section without sharp bends and bottlenecks. The eigenvalue problem is loosely based on the equation for incompressible fluid flow, with some modifications, and is solved by the specialized **Coil Current Calculation** study step.

Add an **Automatic Current Calculation** subnode to the Multi-Turn Coil Domain to set up the automatic computation of the current flow in the coil domain. The boundary conditions for the current calculation study are specified using the Electric Insulation, Input, and Output subnodes available with the node. For information about the use of this study and its functionality, see the *AC/DC Module User's Guide*.

The **Mesh Selection** and **Study Extensions** (Adaptive Mesh Refinement) are described in Common Study Step Settings. There is also detailed information in the Physics and Variables Selection and Values of Dependent Variables sections.

With the AC/DC Module, see Multi-Turn Coil Above an Asymmetric Conductor Plate, model library path **ACDC_Module/Inductive_Devices_and_Coils/multiturn_coil_asymmetric_conductor**.

Electrochemistry Studies and Study Steps

The AC Impedance Stationary, AC Impedance Time Dependent, and Cyclic Voltammetry studies are available with the Battery & Fuel Cells Module, Corrosion Module, Electrochemistry Module, or Electrodeposition Module.

The Time-Dependent with Initialization and Time-Dependent with Initialization, Fixed Geometry studies are available with the Electrodeposition Module or Corrosion Module.

AC IMPEDANCE STATIONARY

The **AC Impedance Stationary** ( ) study is used for electrochemical impedance spectroscopy (EIS) computations in electrochemical cells.

The study consists of two study steps: a Stationary study step followed by a Frequency-Domain, Perturbation study step, which solves for an harmonic linear perturbation of the stationary nonlinear solution. The outputs are Nyquist and Bode plots for selected electrodes over the specified frequency range.

With any of these modules—Battery & Fuel Cells Module, Corrosion Module, Electrochemistry Module, or Electrodeposition Module—see the "Electrochemical Impedance Spectroscopy" model available from the respective model libraries.

AC IMPEDANCE TIME DEPENDENT

The **AC Impedance Time Dependent** ( ) study is used for electrochemical impedance spectroscopy (EIS) computations in electrochemical cells.

The study consists of two study steps: A Time Dependent study step followed by a Frequency-Domain, Perturbation study step, which solves for an harmonic perturbation of the time-dependent solution at the last time step. This study can be used for systems that do not have a steady solution, for example batteries. The outputs are Nyquist and Bode plots for selected electrodes over the specified frequency range.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis i

CYCLIC VOLTAMMETRY

The **Cyclic Voltammetry** () study is used for transient computations of voltammetry experiments together with the Electroanalysis interface.

When this study is added, a **Cyclic Voltammetry** study step is added to the Model Builder. The study step sets up a time-dependent solver. The initial and maximum time step solver settings are based on the settings in the Electroanalysis interface, and a Stop Condition is added to the solver to stop the simulation at the end of the voltammetry cycling. The settings are described for the Time Dependent node.

If the Electroanalysis interface does not contain any Electrode Surface node with active Cyclic Voltammetry settings, no Stop Conditions are added to the solver. Voltammetry simulations can also be performed using a Time Dependent study step.

With any of these modules—Battery & Fuel Cells Module, Corrosion Module, Electrochemistry Module, or Electrodeposition Module—see the “Cyclic Voltammetry at a Macroelectrode in 1D” model available from the respective model libraries.

TIME-DEPENDENT WITH INITIALIZATION

The **Time-Dependent with Initialization** () study can be used to perform transient simulations of electrochemical cells. The study adds a Current Distribution Initialization study step and Time Dependent study step to the study node. The Current Initialization step solves for the electrode and electrolyte potentials as well as all global ODE dependent variables. All other dependent variables in the model, such as concentrations and electrode deformation, are set to the initial values in this step. The Time Dependent step performs a transient simulation for all dependent variables in the model, using the result of the first study step as initial values. See the study steps for settings information.

CURRENT DISTRIBUTION INITIALIZATION

The **Current Distribution Initialization** () study is added to the Time-Dependent with Initialization, Fixed Geometry and Time-Dependent with Initialization studies. You can use this study step as an initialization step for the electric and electrolyte potentials in a transient simulation of an electrochemical cell. It computes a stationary solution for the electrolyte end electric potentials, based on the initial values of the model. The Current Distribution Initialization study step is typically followed by a time-dependent study step that solves for all dependent variables. The settings available for this study step are described for the Stationary node.

With the Electrodeposition Module, see Electrodeposition on a Resistive Patterned Wafer, model library path **Electrodeposition_Module/Tutorial_Models/resistive_wafer**.

TIME-DEPENDENT WITH INITIALIZATION, FIXED GEOMETRY

Use the **Time-Dependent with Initialization, Fixed Geometry** ( ) study to exclude geometry deformation effects from a model. The study is similar to the Time-Dependent with Initialization study, with the difference that the second time-dependent study step does not solve for the geometry deformation dependent variables. This study adds a Current Distribution Initialization study step and Time-Dependent, Fixed Geometry study step to the study node. See the study steps for settings information.

With the Electrodeposition Module, see Electrodeposition on a Resistive Patterned Wafer, model library path **Electrodeposition_Module/Tutorial_Models/resistive_wafer**.

TIME-DEPENDENT, FIXED GEOMETRY

The **Time-Dependent, Fixed Geometry** ( ) study step is added to the Time-Dependent with Initialization, Fixed Geometry study. Use it to exclude the deformation/ALE (X, Y, Z) variables from the variables that are actually solved for by the study step. This is a suitable study step if you want to simulate a time-dependent electrodeposition or corrosion problem for cases where the mesh deformation is expected to be small. The settings available for this study step are described for the Time Dependent node.

Frequency-Domain, Perturbation

The **Frequency-Domain, Perturbation** ( ) study is used for studying small oscillations about a biased solution (small-signal analysis).

With this study and study step, small-signal analysis (AC/DC), prestressed analysis (structural mechanics), or harmonic perturbation (fluid flow) types of analyses can be performed.

When this is added as a study, two study steps are added under the Study node. When it is added as a study step (for example, with the AC Impedance Stationary study type), it is also part of a two-step study that is added under the Study node.

However the study is added to the Model Builder, the first step is usually a **Stationary** study step that computes the stationary (or bias) solution. The second step is this **Frequency-Domain, Perturbation** step, which computes a perturbed solution of the linearized problem around the linearization point (or bias point) computed in the first step. The settings are the same as those for the Stationary and Frequency Domain nodes.

This study (and study step) is available for a variety of interfaces and licenses:

- As a study step with the AC Impedance Stationary and AC Impedance Time Dependent studies, which require the Batteries & Fuel Cells Module or the Electrodeposition Module.
- As a study step for the Small-Signal Analysis, Frequency Domain study, which requires the AC/DC Module or MEMS Module.

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis
- Physics Exclusive and Contributing Node Types

- With the AC/DC Module, see Small-Signal Analysis of an Inductor, model library path **ACDC_Module/Inductive_Devices_and_Coils/small_signal_analysis_of_inductor**.
- With the Batteries & Fuel Cells Module, see Simulation of Electrochemical Impedance Spectroscopy, model library path **Batteries_and_Fuel_Cells_Module/Tutorial_Models/ac_fuel_cell**.
- With the MEMS Module, see Frequency Response of a Biased Resonator—2D, model library path **MEMS_Module/Actuators/biased_resonator_2d_freq**.
- With the Structural Mechanics Module, see Bracket—Frequency Domain Analysis, model library path **Structural_Mechanics_Module/Tutorial_Models/bracket_frequency**.

Frequency-Stationary

The **Frequency-Stationary** ( ) study is used to compute the temperature field, at thermal equilibrium, and the electromagnetic field distribution for models created with the following interface and module combinations:

- Induction Heating, which requires the AC/DC Module,
- Microwave Heating, which requires the RF Module, or
- Laser Heating, which requires the Wave Optics Module.

It is a special case of a Stationary study where the stationary heat transfer equation is solved together with a frequency-domain equation for electromagnetics.

With the RF Module, see RF Heating, model library path **RF_Module/Microwave_Heating/rf_heating**.

Frequency-Transient

The **Frequency-Transient** ( ) study is used to compute temperature changes over time together with the electromagnetic field distribution in the frequency domain.

The study is available with the following interface and module combinations:

- Induction Heating, which requires the AC/DC Module,
- Microwave Heating, which requires the RF Module,
- Laser Heating, which requires the Wave Optics Module.
- Inductively Coupled Plasma, which requires the AC/DC Module and the Plasma Module. For this interface, the temperature represents the electron temperature.
- Microwave Plasma, which requires the RF Module and the Plasma Module. For this interface, the temperature represents the electron temperature.

Physics that support the Frequency Transient study compute electromagnetic fields in the frequency domain and the temperature (electron temperature for Inductively Coupled Plasma and Microwave Plasma) in the time domain. See Time Dependent for all the settings.

Only use this study when the power transfer from the fields to any susceptible variables occurs at twice the angular frequency set by the study. In a large number of cases, the thermal time constant of an object of interest is substantially greater than the angular frequency of the electromagnetic radiation. In order to solve the problem in the time domain, tens or hundreds of thousands of RF cycles need to be computed by the solver before the problem evolves to the periodic steady-state solution. By solving for the fields in the frequency domain, the change in the fields over a single RF cycle does not need to be resolved and thus the periodic steady state solution is reached much more rapidly. This means that the transient, thermal response of an object is computed by this study, but any (small) fluctuations in temperature over any given RF cycle are not.

- With the AC/DC Module, see Inductive Heating of a Copper Cylinder, model library path **ACDC_Module/Electromagnetic_Heating/inductive_heating**.
- With the Plasma Module (plus AC/DC Module), see 3D ICP Reactor, Argon Chemistry, model library path **Plasma_Module/Inductively_Coupled_Plasmas/argon_3d_icp**.
- With the RF Module, see RF Heating, model library path **RF_Module/Microwave_Heating/rf_heating**.

Frozen Rotor

The **Frozen Rotor** ( ) study is used to compute the velocity, pressure, turbulence, concentration, temperature and other fields for flow in rotating machinery and is a special case of a Stationary study. The rotating parts are kept frozen in position, and the rotation is accounted for by the inclusion of centrifugal and Coriolis forces. The study is especially suited for flow in rotating machinery where the topology of the geometry does not change with rotation. It is also used to compute the initial conditions for time-dependent simulations of flow in rotating machinery.

Use the **Frozen Rotor** study with the Rotating Machinery, Laminar Flow and Rotating Machinery, Turbulent Flow interfaces, which require the CFD Module, or the Mixer Module plus the CFD Module. The frozen rotor approach assumes that the flow in the rotating domain, expressed in the rotating coordinate system, is fully developed. See Stationary for all the settings. For information about the use of this study and its functionality, see the *CFD Module User's Guide*.

With the CFD Module: Turbulent Mixing of a Trace Species, model library path **CFD_Module/Single-Phase_Tutorials/turbulent_mixing**.

Frozen Rotor with Initialization

The **Frozen Rotor with Initialization** study is used for simulations of turbulent fluid flow in rotating machinery. It consists of two study steps: a Wall Distance Initialization study step, solving for the distance function to the closest wall, followed by a Frozen Rotor study step solving for velocity, pressure, turbulence and other fields. In the Frozen Rotor study step, the rotating parts are kept frozen in position, and the rotation is accounted for by the inclusion of centrifugal and Coriolis forces.

The study is especially suited for flow in rotating machinery where the topology of the geometry does not change with rotation. It is also used to compute the initial conditions for time-dependent simulations of flow in rotating machinery.

Use the **Frozen Rotor with Initialization** study with the Rotating Machinery, Laminar Flow and Rotating Machinery, Turbulent Flow interfaces, which require the CFD Module, or the Mixer Module plus the CFD Module.

Linear Buckling

The **Linear Buckling** ( ) study is used for estimating the critical load at which a structure becomes unstable.

The Linear Buckling study consists of two study steps: a Stationary study step for applying an external load followed by a **Linear Buckling** study step. In the second study step, an eigenvalue solver is used to compute the buckling modes and the associated critical load factors.

A **Linear Buckling** analysis includes the stiffening effects from stresses coming from nonlinear strain terms. The stiffness coming from stresses and material defines an eigenvalue problem where the eigenvalue is a load factor that, when multiplied with the actual load, gives the critical load in a linear context.

Another way to calculate the critical load is to include large deformation effects and increase the load until the solver fails because the load has reached its critical value.

The Linear Buckling study is available for the Solid Mechanics interfaces using the Structural Mechanics Module or the MEMS Module. It is also available with Shell, Plate, and Truss interfaces when using the Structural Mechanics Module.

STUDY STEP SETTINGS

Use the **Desired number of buckling modes** field to specify the number of buckling modes you want the eigenvalue solver to return.

The **Include geometric nonlinearity** check box and **Mesh Selection** are described in Common Study Step Settings. Also see the Physics and Variables Selection and Values of Dependent Variables sections, which additional detailed information.

With the Structural Mechanics Module: Bracket—Linear Buckling, model library path **Structural_Mechanics_Module/Tutorial_Models/bracket_linear_buckling**.

Mean Energies

The **Mean Energies** ( ) study is used for entering an array of values for the mean electron energy.

COMSOL Multiphysics computes the electron energy distribution function (EEDF) so that the mean electron energy is equal to the mean energy requested. This study conveniently allows data such as rate coefficients, Townsend coefficients, and electron transport properties to be computed as a function of the mean electron energy.

This study is available with the Boltzmann Equation, Two-Term Approximation interface, which requires the Plasma Module. Except for the section below, see Frequency Domain for all settings information. Also see the *Plasma Module User's Guide*.

STUDY SETTINGS

Specify the **Mean energies** to use for the frequency sweep. Enter the mean energies in the field using space-separated numbers or the `range` function.

Use the **Load parameter values** field to select a file with parameter values. You can browse your file system for files by clicking **Browse**. After selecting a file click the **Read File** button to load the parameter values into the **Mean energies** field.

With the Plasma Module: Argon Boltzmann Analysis, model library path **Plasma_Module/Two-term_Boltzmann_Equation/boltzmann_argon**.

Mode Analysis

The **Mode Analysis** ( ) study is used to compute the propagation constants or wave numbers as well as propagating mode shapes, for a given frequency.

For example, in electromagnetics, it is used to compute the propagation constants and mode shapes at ports and waveguide cross sections. In acoustics, it is used to compute the propagation constants and mode shapes at inlets, outlets, and cross sections of guiding structures such as ducts.

When this study is added, it adds a **Mode Analysis** study step under the **Study** node. The Mode Analysis study is available with the Acoustics Module, RF Module, or Wave Optics Module.

STUDY SETTINGS

- Enter a **Desired number of modes**. The default is 6.
- Select a method to **Transform**—**Effective mode index** (the default), **Out-of-plane wave number**, or **None**.
- Enter a value or expression in the **Search for modes around** field. The default is 1.
- Enter a value or expression for the **Mode analysis frequency**. The default is 1 GHz.
- Use the **Mode search method around shift** list to control how the solver searches for modes around the specified value to search around:
  - Select **Closest in absolute value** (the default value) to search for modes that are closest to the search value when measuring the distance as an absolute value.
  - Select **Larger real part** to search for modes with a larger real part than the search value.
  - Select **Smaller real part** to search for modes with a smaller real part than the search value.
  - Select **Larger imaginary part** to search for modes with a larger imaginary part than the search value.
  - Select **Smaller imaginary part** to search for modes with a smaller imaginary part than the search value.

The **Include geometric nonlinearity** check box, **Mesh Selection**, and **Study Extensions** are described in Common Study Step Settings. Also see the Physics and Variables Selection and Values of Dependent Variables sections, which additional detailed information.

- With the Acoustics Module: Eigenmodes in a Muffler, model library path **Acoustics_Module/Industrial_Models/eigenmodes_in_muffler**.
- With the Wave Optics Module: Stress-Optical Effects in a Photonic Waveguide, model library path **Wave_Optics_Module/Waveguides_and_Couplers/stress_optical**.

Optimization

The **Optimization** ( ) study step specifies an optimization problem and controls the optimization solvers provided by the Optimization Module. For information about the use of this study and its functionality, see the *Optimization Module User's Guide*.

Particle Trajectories

The **Particle Trajectories** ( ) study and study step have the same settings as the Time Dependent study step except that by default, only the Particle Tracing Module's interfaces for particle tracing are active in the physics selection.

Prestressed Frequency Analyses Studies

The Prestressed Analysis, Eigenfrequency ( ) and Prestressed Analysis, Frequency Domain ( ) study types perform a stationary analysis of the problem to compute the prestressed state and then perform the appropriate prestressed analysis. In both cases the prestressed state can be computed using any stationary problem.

For exclusive boundary conditions (that is to say, loads which overwrite previously added loads of the same type, such as the potential load in electrostatics), **Harmonic Perturbation** is added as a subnode to the boundary condition node. Its magnitude is added in the subnode itself.

The studies are available for:

- Solid Mechanics using the Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module.
- Electromechanics using the MEMS Module.
- The **Prestressed Analysis, Eigenfrequency** study is also available for the Shell, Plate, and Truss interfaces using the Structural Mechanics Module.

PRESTRESSED ANALYSIS, EIGENFREQUENCY

The **Prestressed Analysis, Eigenfrequency** ( ) study is used to compute eigenfrequencies and eigenmodes that are influenced by a prior static load.

The study consists of two study steps: a Stationary study step followed by an Eigenfrequency study step. The study computes the eigenfrequencies and the shapes of the eigenmodes when influenced by a prior static load on the structure.

The effects of the preload can be computed with or without taking geometric nonlinearity into account. To perform this study, no additional forces need to be added to the physics settings as only the modes and mode frequencies are returned.

- With the MEMS Module: Normal Modes of a Biased Resonator—3D, model library path **MEMS_Module/Actuators/biased_resonator_3d_modes**.
- With the Nonlinear Structural Materials Module: Elastoacoustic Effect in Rail Steel, model library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/rail_steel**.
- With the Structural Mechanics Module: Bracket—Eigenfrequency, model library path **Structural_Mechanics_Module/Tutorial_Models/bracket_eigenfrequency**.

PRESTRESSED ANALYSIS, FREQUENCY DOMAIN

The **Prestressed Analysis, Frequency Domain** ( ) study is used to compute the response to harmonic loads fluctuating around a prior static load.

The study consists of two study steps: a Stationary study step followed by a Frequency Domain study step.

The effects of the preload can be computed with or without taking geometric nonlinearity into account. For this study type it is necessary to specify the magnitude of the harmonic load, as this determines the magnitude of the system response. In order to do this, a **Harmonic Perturbation** force must be added to the model. For contributing loads (that is, loads that can be added without overwriting the same type of node, such as a boundary load in solid mechanics), right-click the node and select **Harmonic Perturbation**. In this case, a load to generate the prestress must be added separately to the model.

For different plot settings made available, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

- With the MEMS Module: Frequency Response of a Biased Resonator—2D, model library path **MEMS_Module/Actuators/biased_resonator_2d_freq**.
- With the Structural Mechanics Module: Bracket—Frequency Domain Analysis, model library path **Structural_Mechanics_Module/Tutorial_Models/bracket_frequency**.

Reduced Electric Fields

Use the **Reduced Electric Fields** ( ) study to sweep through a range of reduced electric fields and to compute electron transport properties and electron impact rate coefficients for a given reduced electric field. Specifying a range of values for the reduced electric field allows for these properties to be tabulated. This tabulated data can then potentially used in a space-dependent plasma simulation. The reduced electric field is defined as the electric field divided by the background gas number density.

This study is available with the Boltzmann Equation, Two-Term Approximation interface, which requires the Plasma Module. Except for the section below, see Frequency Domain for all settings. Also see the *Plasma Module User's Guide*.

STUDY SETTINGS

Specify the **Reduced electric fields** to use for the frequency sweep. Enter the reduced electric fields in the field using space-separated numbers or the `range` function.

Use the **Load parameter values** field to select a file with parameter values. You can browse your file system for files by clicking **Browse**. After selecting a file click the **Read File** button to load the parameter values into the **Reduced electric fields** field.

With the Plasma Module: Argon Boltzmann Analysis, model library path **Plasma_Module/Two-term_Boltzmann_Equation/boltzmann_argon**.

Small-Signal Analysis, Frequency Domain

The **Small-Signal Analysis, Frequency Domain** ( ) study is used for studying small oscillations about a biased solution in electromagnetics.

The study consists of two study steps: a Stationary study step, for computing the biased solution, followed by a Frequency-Domain, Perturbation study step, for computing the frequency response about the biased solution. For the second study step, the computation is for a linear perturbation about the biased solution.

This study is available with the AC/DC Module, MEMS Module, or Semiconductor Module.

> With the AC/DC Module: Small-Signal Analysis of an Inductor, model library path **ACDC_Module/Inductive_Devices_and_Coils/small_signal_analysis_of_inductor**.

Stationary and Time Dependent One-Way Coupled Studies for Fluid Structure Interaction

The Stationary, One-Way Coupled, Time Dependent, One-Way Coupled, Stationary, One-Way Coupled with Initialization, and Transient, One-Way Coupled with Initialization studies are available with the Fluid-Structure Interaction interface, which requires the MEMS Module or the Structural Mechanics Module.

The initialization study types are also available if you have a Structural Mechanics Module *plus* the CFD Module.

STATIONARY, ONE-WAY COUPLED

The **Stationary, One-Way Coupled** ( ) study is used for stationary fluid-structure interaction computations where the model is one-way coupled in the sense that the structural deformations are so small that they do not affect the flow.

The study consists of two study steps: a Stationary, Fluid study step, solving for the fluid velocity and pressure fields, followed by a Stationary, Solid study step, solving for the solid deformation field. The settings available for the two study steps are the same as for a Stationary study step. If an additional physics interface is added, then it is automatically added to both study steps.

TIME DEPENDENT, ONE-WAY COUPLED

The **Time Dependent, One-Way Coupled** ( ) study is used for time-dependent fluid-structure interaction computations where the model is one-way coupled in the sense that the structural deformations are so small that they do not affect the flow.

The study consists of two study steps: a Time Dependent, Fluid study step, solving for the fluid velocity and pressure fields, followed by a Time Dependent, Solid study step, solving for the solid deformation field. The settings available for the two study steps are the same as for a Time Dependent study step. If an additional physics interface is added, then it is automatically added to both study steps.

STATIONARY, ONE-WAY COUPLED WITH INITIALIZATION

The **Stationary, One-Way Coupled with Initialization** ( ) study is used for stationary turbulent fluid-structure interaction computations where the model is one-way coupled in the sense that the structural deformations are so small that they do not affect the flow.

The study consists of three study steps: a Wall Distance Initialization study step, solving for the distance function to the closest wall, followed by a Stationary, Fluid study step, solving for the fluid field, and finally a Stationary, Solid study step, solving for the solid deformation field. The settings available for the study steps are the same as for a Stationary study step. If an additional physics interface is added, then it is automatically added to the last two study steps.

How to Add this Study

1 Add a **Fluid-Structure Interaction** interface to the model.

2 On the **Fluid-Structure Interaction** settings window under **Physical Model**, select **RANS** as the **Turbulence model type**.

3 Select **Low Reynolds number k-ε** or **Spalart-Allmaras** as the **Turbulence model**.

4 In the **Model Builder**, right-click the root node and select **Add study**.

5 In the **Model Wizard** or **Add Study** window under **Preset Studies**, select the **Stationary, One-Way Coupled with Initialization** study.

TRANSIENT, ONE-WAY COUPLED WITH INITIALIZATION

The **Transient, One-Way Coupled with Initialization** ( ) study is for time-dependent, one-way coupled Fluid-Structure Interaction models using a turbulence model that requires the distance to the closest wall. The study node creates three study steps:

- The first, Wall Distance Initialization solves for the distance to the closest wall.
- The second, Time Dependent, Fluid solves for the fluid flow variables.
- The third, Time Dependent, Solid solves for the solid deformation.

The settings available for the study steps are the same as for the Time Dependent node. When additional physics is included, it is by default added to the last two study steps.

How to Add this Study

1 Add a **Fluid-Structure Interaction** interface to the model.

2 On the **Fluid-Structure Interaction** settings window under **Physical Model**, select **RANS** as the **Turbulence model type**.

3 Select **Low Reynolds number k-ε** or **Spalart-Allmaras** as the **Turbulence model**.

4 In the **Model Builder**, right-click the root node and select **Add study**.

5 In the **Model Wizard** or **Add Study** window under **Preset Studies**, select the **Transient, One-Way Coupled with Initialization** study.

STATIONARY, FLUID

The **Stationary, Fluid** ( ) study step is added to the Stationary, One-Way Coupled or Stationary, One-Way Coupled with Initialization study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the Stationary node.

TIME DEPENDENT, FLUID

The **Time Dependent, Fluid** ( ) study step is added to the Time Dependent, One-Way Coupled or Transient, One-Way Coupled with Initialization study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the Time Dependent node.

STATIONARY, SOLID

The **Stationary, Solid** ( ) study step is added to the Stationary, One-Way Coupled Stationary, One-Way Coupled with Initialization study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the Stationary node.

TIME DEPENDENT, SOLID

The **Time Dependent, Solid** ( ) study step is added to the Time Dependent, One-Way Coupled or Transient, One-Way Coupled with Initialization study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the Time Dependent node.

Stationary Plug Flow

The **Stationary Plug Flow** ( ) study is used for plug flow reactor models created with the Reaction Engineering interface. The reactor equations describe the molar flow rate (SI unit: mol/s) as a function of reactor volume (SI unit: m^3) under stationary conditions. Selecting the Reaction Engineering interface and the Stationary Plug Flow study sets up a plug flow reactor type by default. The settings for this study are described for the Time Dependent node. This study requires the Chemical Reaction Engineering Module.

With the Chemical Reaction Engineering Module: Nonisothermal Plug Flow Reactor, model library path **Chemical_Reaction_Engineering_Module/Tubular_Reactors/ nonisothermal_plug_flow**.

Stationary with Initialization and Transient with Initialization

The Stationary with Initialization study is available with a variety of physics interfaces, which in general require the CFD Module or Heat Transfer Module.

The Transient with Initialization study is available with a many physics interfaces and module combinations. In general this study requires a CFD Module, Heat Transfer Module, or Microfluidics Module.

Whether a Wall Distance Initialization or Phase Initialization study step is added is based on whether it is a turbulent flow or two-phase flow model.

STATIONARY WITH INITIALIZATION

The **Stationary with Initialization** ( ) study is used for stationary two-phase flow and turbulent flow models that require an initialization step.

Turbulent Flow Interfaces

For stationary turbulent flow models that require an initialization, it adds a Wall Distance Initialization study step followed by a Stationary study step to the Study node. The **Wall Distance Initialization** study step is dedicated to solving for the reciprocal wall distance, that is, the reciprocal distance to the closest wall. The second step is an ordinary **Stationary** study step, but it excludes the reciprocal wall distance except when the physics is put on a moving frame, in which case the reciprocal wall distance is solved for in the stationary step as well.

If you have the CFD Module, see an example using the:

- Turbulent Flow, SST interface: Flow Around an Inclined NACA 0012 Airfoil, model library path **CFD_Module/Single-Phase_Benchmarks/naca0012_airfoil**.
- High Mach Number Flow interface: Transonic Flow in a Sajben Diffuser, model library path **CFD_Module/High_Mach_Number_Flow/sajben_diffuser**.

Two-Phase Flow Interfaces and Moving Interfaces

For stationary two-phase flow models that require an initialization of a level set function or phase field function, it adds a Phase Initialization study step followed by a Stationary study step to the Study node. The **Phase Initialization** study step is dedicated to solving for the reciprocal distance to the phase interface. The second step is an ordinary **Stationary** study step, but it excludes the reciprocal interface distance.

TRANSIENT WITH INITIALIZATION

The **Transient with Initialization** ( ) study is used for time-dependent two-phase flow and turbulent flow models that require an initialization step.

Turbulent Flow Interfaces

For time-dependent turbulent flow models that require an initialization, it adds a Wall Distance Initialization study step followed by an ordinary Time Dependent study step to the Study node. The **Wall Distance Initialization** study step is dedicated to solving for the reciprocal wall distance, that is, the reciprocal distance to the closest wall.

Two-Phase Flow Interfaces and Moving Interfaces

For time-dependent two-phase flow models that require an initialization of a level set function or phase field function, it adds a Phase Initialization study step followed by an ordinary Time Dependent study step to the Study node. The **Phase Initialization** study step is dedicated to solving for the reciprocal distance to the phase interface.

See an example using the Laminar Two-Phase Flow, Level Set interface:

- If you have the CFD Module: Droplet Breakup in a T-Junction, model library path **CFD_Module/Multiphase_Benchmarks/droplet_breakup**.
- If you have the Microfluidics Module: Droplet Breakup in a T-Junction, model library path **Microfluidics_Module/Multiphysics_Module/Two-Phase_Flow/droplet_breakup**.

WALL DISTANCE INITIALIZATION

The **Wall Distance Initialization** ( ) study step is added to the Stationary with Initialization and Transient with Initialization studies when using a turbulent model. This first step is dedicated to solving for the reciprocal wall distance. For turbulence models, the distance determined in the initialization step is the distance to the closest wall. For two-phase flow it is the distance to the phase interface. The settings for this study are the same as for the Stationary and Time Dependent studies.

PHASE INITIALIZATION

The **Phase Initialization** ( ) study step is added to the Stationary with Initialization and Transient with Initialization studies when using two-phase flow or a moving interface. This first step is dedicated to solving for the reciprocal interface distance. In the case of two-phase flow the distance determined in the initialization step is the distance to the phase interface. The settings for this study are the same as for the Stationary and Time Dependent studies.

Computing a Solution

A solution can be computed in a few ways depending on the sequence implemented for a model.

THE DIFFERENCE BETWEEN COMPUTING A STUDY, SOLVER CONFIGURATION, OR JOB CONFIGURATION

There are conceptual differences between computing, or running, a study versus running a solver (configuration) or a job (configuration). When a solver configuration or job configuration is run, no nodes are added or removed; it is computed “as is.”

Running a study, on the other hand, can be different. When you compute a Study, it always runs the enabled configurations (see below). What the Study actually runs can be a Job or a Solver (for example, a Stationary Solver) depending on the study configuration. If a Job is run, it typically also means that a a solver is also run (by the job). But before the study runs a job or a solver, it reconstructs the (enabled) configurations from scratch. An exception to this rule is when the enabled configurations are edited (an asterisk indicates this, see Figure 19-7 for an example), in which case the sequences is computed “as is.”

ABOUT ENABLED STUDY OR SOLVER SEQUENCES

The particular sequence that is **Enabled** and runs when selecting **Compute** has a green border around its icon ( ). You can disable an enabled sequence by right-clicking the node and selecting **Disable** (which removes the green border). If no sequence is enabled when the study or solver configuration attempts to generate a sequence, a new sequence with default settings is generated. Only one sequence per study can be enabled. Also see Figure 19-6 for other examples of enabled sequences.

COMPUTING A STUDY

The most straightforward method to compute a solution is to right-click the Study node ( ) and select **Compute** ( ) (or press F8).

By default, a study creates a Solution data set and plot groups with results plots suitable for the physics that you compute the solution for. If you do not want to generate plots automatically, clear the **Generate default plots** check box in the **Study Settings** section in the main Study node settings window.

If the study contains more than one study step, and you want to compute only a part of the study steps, right-click and select:

- **Compute Selected Step** ( ) to compute just that study step, or
- **Compute to Selected** ( ) to compute for all study steps except the first one, or
- **Compute from Selected** ( ) to compute for all study steps except the last one.

UPDATING A SOLUTION

To update the solution for a particular study, right-click the **Study** node and select **Update Solution** ( ) (or press F5).

Updating the solution updates the current study (if selected) or all studies if no study is selected. This is useful in the following situations when you have:

- Added or edited variables and want to use these during analysis without having to solve the model again.
- Edited the equations, material properties, or boundary conditions and want to use these without having to solve the model again.

- Changed the element order and want to interpolate the solution onto the new elements for results analysis or other purposes.
- Remeshed or modified a geometry and want to interpolate the solution onto the new geometry for results analysis or other purposes.

In all these cases the software passes or interpolates the solution to the resulting data sets but does not recompute it to reflect any changes in variables, equations, mesh, or geometry.

COMPUTING A SOLVER CONFIGURATION

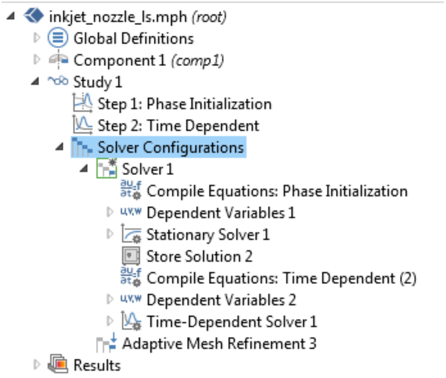
When you have added study steps to a study, a Solver Configurations (and maybe a Job Configurations) sequence is generated when the Study is computed. The Solver Configurations branch represents the solvers, dependent variables, and other study-related functionality that the study steps require.

> A Solver Configurations or Job Configurations node displays automatically if it has content. Otherwise, if you click the **Show** button ( ) and select **Advanced Study Options** it is available as an option from the context menu.

In some cases, the default settings in the study steps are not sufficient to specify the details of how to obtain a solution. In this case you can edit the sequence and run it again. See Editing and Re-Running a Solver Configuration below.

To compute a solution:

- Under **Solver Configurations**, right-click the corresponding **Solver** node and select **Compute**.
- Right-click a Study node and select **Compute** to compute the enabled solver (the node with a green border around its icon), if such a solver configuration exists. If no solver configuration exists, or if all sequences are disabled, a new solver configuration is generated and computed.



```
inkjet_nozzle_ls.mph (root)
  Global Definitions
  Component 1 (comp1)
  Study 1
    Step 1: Phase Initialization
    Step 2: Time Dependent
    Solver Configurations
      Solver 1
        Compile Equations: Phase Initialization
        Dependent Variables 1
        Stationary Solver 1
        Store Solution 2
        Compile Equations: Time Dependent (2)
        Dependent Variables 2
        Time-Dependent Solver 1
      Adaptive Mesh Refinement 3
  Results
```

Figure 19-7: A Solver Configurations sequence with more than one solver. This is an example from the CFD Module model library.

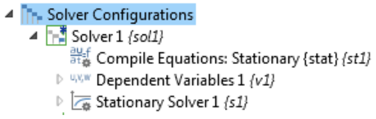
If you have already generated a solver configuration for the study, or if your solver configuration consists of several solvers, as with the study steps you can right-click a configuration and choose:

- **Compute to Selected** ( ) (or press F7), to run a particular solver, or
- **Compute from Selected** ( ) to run the selected solver and all solvers below it in the sequence, or
- **Compute** (or press F8), to run the entire solver configuration.

For example, right-click a Dependent Variables node and select **Compute to Selected** to evaluate the initial values for the dependent variables (similar to the **Get Initial Value** and **Get Initial Value for Step** options for the main **Study** nodes and the study steps).

EDITING AND RE-RUNNING A SOLVER CONFIGURATION

The Solver Configurations branch nodes (or if applicable, the Job Configurations branch) can be edited to adjust solver settings. For example, if you want to change a tolerance or use a different time-stepping method. If you edit any settings in a subnode to a **Solver** node, an asterisk in the upper right corner (Figure 19-8) indicates that the settings differ from the default settings for the study types in the study.



Solver Configurations
Solver 1 *{sol1}*
Compile Equations: Stationary {stat} *{st1}*
Dependent Variables 1 *{v1}*
Stationary Solver 1 *{s1}*

Figure 19-8: A Solver node has an asterisk in the icon to indicate the sequences that have been edited. To compute, highlight the Solver node, press F8, right-click and choose Compute, or click the button on the settings window.

After editing a solver configuration, you run the sequence again. Running a solver configuration is tantamount to computing a solution. Like mesh nodes, solution nodes are not built automatically as they are added. It is possible to have several solver configurations under a Solver Configurations node (see Figure 19-7).

PROGRESS AND LOG INFORMATION

While a problem is being solved it is useful to know its progress. The Progress Window monitors the state of the analysis for the solvers during the solution process. In this window you can Cancel or Stop a Solver Process. Alternatively, in The Log Window you can inspect convergence information and other data from the latest and earlier runs.

- Getting Results While Solving and Convergence Plots
- Computing the Initial Values

Getting Results While Solving

The ability to get numerical results and plots of the intermediate solutions while solving can be useful for diagnosing problems and for monitoring the progress of the solution. COMSOL Multiphysics supports two ways of displaying results while solving:

- Plots in The Graphics Window: you can select any of the plot groups in the model to display while solving.
- Probe data in the **Table** window (see The Table Window and Tables Node) and associated line graphics in a separate **Probe Plot** window: you can include any probe defined in the model. See Probes.
- In Results While Solving sections on the study step settings windows you can control which plots to display while solving. See Common Study Step Settings.

Computing the Initial Values

In some situations you might want to evaluate and plot values, expressions, or functions that need not be solved for, such as initial values and functions evaluated using the initial values.

To make the initial values available for results evaluation and plotting, right-click the Study node ( ), and select **Get Initial Value** (U t=0). By default this plots the initial values of the variables solved for as specified by the **Field** subnode (W U.T.P) under a Dependent Variables node for the first study step in the solver configuration.

For each study step you can also right-click and choose **Get Initial Value for Step**) to evaluate the initial value for that step.

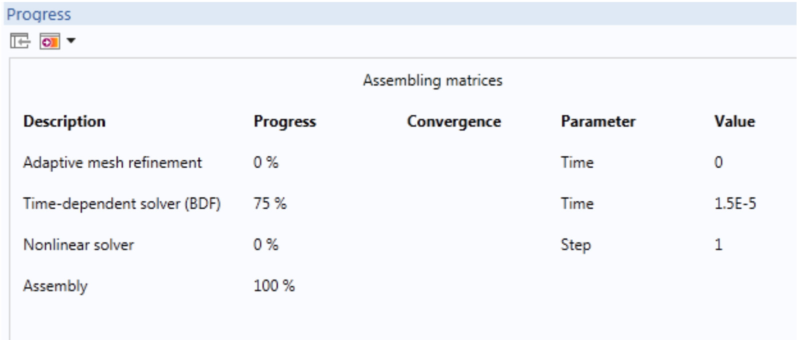
Also variables not solved for (such as a solution from a previous time-dependent or parametric analysis) can be made available for results evaluation and visualization. Select **Values of variables not solved for** from the **Keep solution** list under **Output** in the **Dependent Variables** node.

The Progress Window

The **Progress** window ( ) displays the progress of the solver or mesher during the process, including a progress bar and progress information for each solver or mesher. You can view the progress from the status bar in the lower-right corner of the COMSOL Desktop.

The solvers call each other in a hierarchical order: the adaptive solver calls the linear, nonlinear, parametric, eigenvalue, or time-dependent solver; the parametric solver calls the nonlinear or linear solver; the time-dependent, eigenvalue, linear, and nonlinear solvers all call both the assembly and the linear system solver. The solver hierarchy is visible in the Progress window because each solver adds its own line when it is called.

This window is always available but is empty when no progress information is available. For a log of the progress information, see The Log Window.



Assembling matrices

Description	Progress	Convergence	Parameter	Value
Adaptive mesh refinement	0 %		Time	0
Time-dependent solver (BDF)	75 %		Time	1.5E-5
Nonlinear solver	0 %		Step	1
Assembly	100 %			

Figure 19-9: An example of the Progress Window solving a model.

The **Progress** window has the following columns:

- The **Description** column shows the solver name.
- The **Progress** column shows an estimate of the solver progress.
- The **Convergence** column shows an estimate of the solver's convergence if available. Also see Convergence Plots for more information.
- The **Parameter** and **Value** columns contain solver-dependent information: the adaptive solver shows the adaptive mesh generation number; the time-dependent solver shows the time; the parametric solver shows the parameter value; and the nonlinear solver and iterative linear system solvers show the iteration number.

If you want to continue an interrupted parametric or time-dependent solution, for example, right-click the **Study** node for a predefined study type and select **Continue**).

CANCEL OR STOP A SOLVER PROCESS

You can cancel or stop the solver process if the model's solving time or likelihood for convergence is not progressing as expected. Use the horizontal or vertical scrollbars if needed. On the status bar you can follow the progress, cancel the solver process by clicking the **Cancel** button ( ), or click the **Progress** button ( ) to open this window.

You can use the **Stop** buttons that appear at each solver level to stop the solver's execution. When you click a **Stop** button, COMSOL Multiphysics returns a current approximation to the solution if possible. For example, when you click it at the adaptive solver level, the underlying linear, nonlinear, parametric, eigenvalue, or time-dependent solver continues until it is finished, but the adaptive solver stops at its current generation, immediately returning a solution. Similarly, you can click the **Stop** button to return the current iteration for the nonlinear solver or an iterative solver. Also use the **Stop** button during time stepping to return all time steps up to the current one. The parametric solver works similarly: to return the solutions for all parameter values up to the current one, click the **Stop** button.

- Meshing
- Convergence Plots

Convergence Plots

Convergence plots use graphics to show how an error estimate or time step evolves during the solution process for nonlinear, time dependent, and parametric solvers. By default convergence plots are generated.

To control which solvers generate convergence plots, click the **Convergence Plot Settings** ( ) button on The Progress Window toolbar to select or clear the convergence plots for each solver. For example, for a nonlinear time-dependent model, the menu that contains the nonlinear solver and Time-Dependent Solver (Generalized-alpha) as options.

By default, all solvers are selected, and the convergence plot for each solver appears in its own **Convergence Plot** window. Click to clear the check mark for a solver to turn off its convergence plot.

Click the **Copy Convergence Data to Model** button ( ) on the toolbar to copy the convergence data to a table in the **Table** window.

NaNs (Not-a-Number entries) that appear in the tabulated convergence data represent breaks between multiple convergence cycles in an interactive solution process.

CONVERGENCE INFORMATION IN THE PLOTS

The convergence plots show an error estimate against the iteration number for the nonlinear solver and for the iterative linear system solvers (the Conjugate gradients, BiCGStab, GMRES, FGMRES, and multigrid solvers). See Convergence Criteria for Iterative Solvers.

For the nonlinear solver, the convergence plots show the relative error estimate for each nonlinear iteration number. This number also appears in The Log Window as **ErrEst**. The segregated solver shows one plot with one graph for each segregated step.

For the iterative linear system solvers, the error estimate for each linear iteration is a factor times the relative (preconditioned) residual. This number also appears in the Log window as **LinErr**. When these solvers are used together with the nonlinear solver, the graphs for the different linear-system solution steps are merged, and the plots use the accumulated number of iterations. Each linear solver used has a separate plot window.

When using the parametric solver, the graphs for the different parameter steps are merged, and the convergence plots use the accumulated number of iterations. The graphs for the different nonlinear and linear solve steps are concatenated. The plot uses the accumulated number of iterations.

When using a Time-Dependent Solver, the graph in the **Convergence Plot** window shows the reciprocal of the time step size versus the time step. That is, a convergence plot with decreasing values shows that the time-dependent solver takes longer time steps, and vice versa.

The error estimate numbers for the last iteration also appear in the **Convergence** column in The Progress Window.

CHANGING THE DEFAULT SETTINGS

Open The Preferences Dialog Box and click **Results** to edit the preferences for the plots that you can use to monitor solver convergence.

- The **Generate convergence plots** check box is selected by default. Clear that check box if you do not want the software to generate convergence plots.
- To control the size of the buffer used for storing the steps in the convergence plot, in the **Convergence plot buffer size (steps per plot)** field (default value: 10,000 steps).

The Log Window

The **Log** window (▦) contains information from previous solver runs, including convergence information, solution time, and memory use. When a solver starts, the log window displays logs from all solver calls. This window is always available. For progress information during a solver or mesher process, see The Progress Window.

A horizontal divider (`===============`) indicates the start of a new solver progress log. To differentiate logs from different models, the log contains a horizontal divider displaying the name of the Model MPH-file each time a model is opened. For example,

```
================ Opened thin_layer_diffusion.mph =================
```

It also contains a similar divide when you save a model to a new file (using **Save As**):

```
================= Saved thin_layer_diffusion.mph =================
```

When a solver starts working, it prints the number of degrees of freedom in the linear systems to be solved to the log. For certain problems, there are additional degrees of freedom involved in the discrete problem formulation that do not affect the size of the matrices assembled by the solver. These are called *internal degrees of freedom* and are displayed separately from the actual degrees of freedom in the log. For example, when solving plasticity problems in structural mechanics, plastic strains are represented by internal degrees of freedom.

When a solver has finished it reports the following information:

- The solution time (in seconds)
- The maximum amount of physical memory used (in MB)
- The maximum amount of virtual memory used (in MB)

In addition, the log includes the following information that is specific to the type of solver:

- The Adaptive and Parametric Solver Logs
- The Nonlinear Solver Log
- The Iterative Linear System Solver Logs
- The Time-Dependent Solver Log
- The Eigenvalue Solver Log
- The Optimization Solver Log

Scrolling in the Log Window

You can scroll the contents of the **Log** window to display information from earlier runs.

- Click the **Scroll Lock** button (🔒) to stop the window from scrolling the log during a solver call, for example.
- Click the **Scroll Lock** button again to resume scrolling.
- Click the **Clear** button (🧹) to clear the **Log** window from all information.

Buffer Size of the Log Window

By default, the buffer size of the **Log** window is limited to 300,000 characters.

To change the size of the buffer, go to The Preferences Dialog Box **General** section and then enter a maximum buffer size (in characters) in the **Log window size (characters)** field. The default is 300,000 characters. This buffer size also applies to the Log stored in the solvers for the last run.

THE ADAPTIVE AND PARAMETRIC SOLVER LOGS

The Adaptive Mesh Refinement solver prints a section in the log for each adaptive generation containing the current number of elements and an error indicator. The Parametric solver similarly outputs one section to the log for each parameter value.

The Stationary Solver, Time-Dependent Solver, and Eigenvalue Solver log their iterations.

THE NONLINEAR SOLVER LOG

The log from the nonlinear solver contains the following information:

- The iteration number (**Iter**).
- A relative error estimate (**ErrEst**).
- The damping factor used in each Newton step (**Damping**).
- Fraction of Newton and Cauchy steps for the Double dogleg solver (**Newton, Cauchy**).
- The size of the undamped Newton step (**Stepsize**) in the error estimate norm.
- The numbers of residuals (**#Res**), Jacobians (**#Jac**), and linear-system solutions computed (**#Sol**) so far.

THE DIRECT LINEAR SYSTEM SOLVER LOGS

The direct linear system solvers produce a log that additionally contains a relative error estimate (**LinErr**) and the relative residual (**LinRes**). The relative error is estimated by deferred correction (also called iterative improvement)—that is, by solving $A{\cdot}dx = r(x)$ for dx and setting $\mathrm{LinErr} = \mathrm{rhoB}\cdot\mathrm{norm}(dx)/\mathrm{norm}(x)$, where rhoB is the *factor in error estimate* value from the direct solver, and $r(x) = Ax - b$. The relative residual is the Euclidean norm of the residual divided by the norm of the linear system's right-hand side; that is, $\mathrm{LinRes} = \mathrm{rhoB}\cdot\mathrm{norm}(r(x))/\mathrm{norm}(b)$.

THE ITERATIVE LINEAR SYSTEM SOLVER LOGS

The iterative linear system solvers produce a log that additionally contains the total number of linear iterations (**LinIt**), a relative error estimate (**LinErr**), and the relative residual (**LinRes**). The relative error estimate is a factor times the relative (preconditioned) residual. The relative residual is the Euclidean norm of the residual divided by the norm, $|b|$, of the linear system's right-hand side.

Convergence Criteria for Iterative Solvers

THE TIME-DEPENDENT SOLVER LOG

For the Time-Dependent Solver, the time-stepping algorithm produces a log that contains:

- The time step number (**Step**).
- Time (**Time**; output times are indicated with **out**).
- The step size (**Stepsize**).
- The number of residuals (**Res**), Jacobians (**Jac**), and linear system solutions (**Sol**) computed.

You can see also the order of accuracy of the method (**Order**), the number of error test failures in time stepping (**Tfail**), and the number of failures in the nonlinear iterations (**NLfail**). For iterative linear system solvers, the log also contains the total number of linear iterations, a linear error estimate, and a relative residual (see above).

THE EIGENVALUE SOLVER LOG

The Eigenvalue Solver produces a log that contains the iteration number (**Iter**), an error estimate (**ErrEst**), the number of converged eigenvalues (**Nconv**), and— if you are using an iterative linear solver— the number of linear iterations (**LinIt**).

THE OPTIMIZATION SOLVER LOG

SNOPT

The Optimization Solver (which requires the Optimization Module) algorithm used by the solver SNOPT is an iterative procedure that involves *major* and *minor* iterations. A major iteration results in a new solution candidate. For each major iteration, the optimization solver solves a quadratic-programming subproblem using an iterative procedure; these iterations are the minor iterations.

The log produced by the optimization solver SNOPT has the following data:

- The cumulative number of minor iterations (**Itns**).
- The current major iteration number (**Major**).
- The number of minor iterations for the current major iteration (**Minor**). This value should be 1 when the solver is close to the solution.
- The step length taken in the current search direction (**Step**). This value should be 1 when the solver is close to the solution.
- The number of times the multiphysics model has been solved (**nPDE**).
- The maximum complementarity gap (**Error**). It is an estimate of the degree of nonoptimality of the reduced costs. For convergence, this value should be smaller than the **Optimality tolerance**.
- The current value of the objective function (**Objective**).

MMA

The MMA solver implements another general-purpose optimization algorithm. The method is based on solving a sequence of approximating subproblems, one for each *inner* iteration. The subproblem is constructed from function values and gradients, which are evaluated once per *outer* iteration only. Each outer iteration requires one or more *inner* iterations, depending on whether the last subproblem was found to be conservative or not. Once a feasible point is found, outer iterates stay feasible. If the initial guess is infeasible or the feasible set is empty, nonzero infeasibilities may be reported.

The log produced by the MMA solver contains the following data:

- The cumulative number of outer iterations (**Iter**). One outer iteration per line is reported in the log.
- The number of inner iterations for the current outer iteration (**Inner**). This is the number of attempts needed to find a conservative approximating subproblem, which in some sense measures the nonlinearity of the problem.

- The cumulative number of model evaluations (**nPDE**). Each inner iteration requires a model evaluation in order to check the conservativeness of the approximation. Gradients are only computed once for each outer iteration.
- The estimated error (**Error**). The error is defined as the maximum relative change in any control variable since last outer iteration, computed as a percentage of the distance between the control variable's bounds.
- The current value of the objective function (**Objective**).
- The maximum violation of any constraint (**MaxInfeas**). For a feasible solution, this number must be zero. It may be nonzero in the first outer iterations if the initial guess is infeasible.

Levenberg-Marquardt

The optimization solver Levenberg-Marquardt is an iterative procedure used to solve least-squares problems. The log produced by the Levenberg-Marquardt solver contains the following data:

- The number of Levenberg-Marquardt iterations (**Itns**).
- The current Levenberg-Marquardt factor (**lmFact**). A small factor typically indicates fast convergence.
- The number of times the multiphysics model has been solved (**nPDE**).
- The maximum absolute value of the gradient (with respect to the control variables) of the objective function (**Gradient**).
- The estimated error based on the gradient, the objective function, and the control variables (**Error**). For convergence, this value should be smaller than the **Optimality tolerance**.
- The current value of the objective function (**Objective**).

The External Process Window

Use the **External Process** window ( ) to follow external processes (such as distributed batch jobs) that have been started. The window updates when you are attached to the external process. You can do operations that are performed in the External Process nodes under a Parametric or Batch job configuration by selecting an external process from the list and then select the operation. When detached, you can reattach by pressing the **Attach Job** button. The window opens automatically when you start a batch process. See Table 19-4 for descriptions of the toolbar buttons available on this window.

To open the **External Process** window:

- Windows users, from the **Home** ribbon select **More Windows>External Process**.
- Cross platform (Mac and Linux) users, select **Windows>External Process**.

TABLE 19-4: EXTERNAL PROCESS TOOLBAR BUTTONS

ICON	NAME	DESCRIPTION
	Attach Job	If a job has been detached, click to reattach the job to run external processes and follow the status updates.
STOP	Stop All Processes	Sends the stop command to unfinished jobs. Similar to using the Stop button for The Progress Window.
⊗	Cancel All Processes	Sends the cancel command to unfinished jobs. Similar to using the Cancel button for The Progress Window.
STOP	Stop Process	Stops the selected process. Similar to using the Stop button for The Progress Window
⊗	Cancel Process	Cancels the selected process. Similar to using the Cancel button for The Progress Window.
	Rerun Job	Restarts the selected job.

TABLE 19-4: EXTERNAL PROCESS TOOLBAR BUTTONS

ICON	NAME	DESCRIPTION
	Clear Status	Clears the status of the selected job. Useful when the status indicates that the process is running but the process has failed.
	Log	Shows the current log of the selected process.
	Open File	Opens the output file from the selected process.
	Batch Jobs (generated tables)	Selects the Batch Jobs to view.

- Studies: Batch, Cluster Sweep, Batch Sweep, and Cluster Computing
- Batch (Job Configuration)
- External Class

Solution Operation Nodes and Solvers

The first few sections provide some background information about the solvers and the algorithms used:

- Selecting a Stationary, Time-Dependent, or Eigenvalue Solver
- Remarks on Solver-Related Model Characteristics
- Scaling of Variables and Equations
- About the Stationary Solver
- About the Parametric Solver
- About the Time-Dependent Solver
- About the Time Discrete Solver
- The Eigenvalue Solver Algorithm
- The Modal Solver Algorithm
- The Time Explicit Solver Algorithms

About Solver Commands in the *COMSOL API Reference Manual*

Then the settings for the solver operation nodes listed in Table 19-5 are detailed. There is also a list of the References for the Solution Operation Nodes and Solvers.

TABLE 19-5: SOLUTION OPERATION NODES

ICON	NAME	DESCRIPTION
	AWE Solver	Solve a parametric problem with asymptotic waveform evaluation.
u,v,w	Dependent Variables	Handles the dependent variables solved for (initial values, scaling) and dependent variables not solved for (prescribed values).
	Eigenvalue Solver	Solve linear or linearized eigenvalue problems (also called eigenfrequency problems). Also see The Eigenvalue Solver Algorithm.
	Modal Solver	To solve either parameter stepping (also called frequency response) or time stepping (also called transient response) problems using a reduced model. Also see The Modal Solver Algorithm.
	Optimization Solver	Solve PDE-constrained optimization problems. Requires the Optimization Module.
	Plug Flow Solver	Solve a plug flow reactor model (requires the Chemical Reaction Engineering Module).
	Stationary Solver	Solve linear and nonlinear stationary problems (also called static or steady-state problems). Also see About the Stationary Solver.
	Time-Dependent Solver	Solve time-dependent problems (also called dynamic or unsteady problems) using the implicit time-stepping methods: BDF or generalized-α. Also see About the Time-Dependent Solver.

TABLE 19-5: SOLUTION OPERATION NODES

ICON	NAME	DESCRIPTION
	Time Discrete Solver	Solve time-dependent problems (dynamic or unsteady problems) that have already been discretized in time using, for example, the prev operator or the bdf operator.
	Time Explicit Solver	Solve time-dependent problems (also called dynamic or unsteady problems) using the family of Runge-Kutta explicit time-stepping schemes or the Adams-Bashforth 3 solver. Also see The Time Explicit Solver Algorithms.

Selecting a Stationary, Time-Dependent, or Eigenvalue Solver

The chosen study type adds the appropriate solvers for the study; you do not need to select one yourself. If you prefer to make a selection, the first question to ask is whether the problem is stationary or time-dependent.

Most real-world phenomena develop in time, but you might know that the system under study approaches a *steady state* described by a stationary solution.

For a stationary problem, select the Stationary Solver. When solving the time-dependent coefficient form problem

$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au = f$$

the stationary solver searches for a solution where $\partial u/\partial t = 0$. The Time-Dependent Solver and the stationary solver handle linear as well as nonlinear problems.

In some cases you might want to study the natural harmonic oscillations of a time-dependent problem. This involves finding the eigensolutions u with the associated eigenvalues λ in a PDE problem of the following form:

$$\nabla\cdot(-c\nabla u-\alpha u)+\beta\cdot\nabla u+au = \lambda d_a u-\lambda^2 e_a u$$

Such an analysis is particularly interesting in electromagnetics, structural mechanics, acoustics, and wave propagation. To study the eigensolutions and compute the eigenvalues (or eigenfrequencies), select the Eigenvalue Solver.

In addition to these fundamental solvers, COMSOL Multiphysics includes additional solvers for special applications such as an Optimization Solver (which requires the Optimization Module), a Time Discrete Solver, an AWE Solver, and a Plug Flow Solver (which requires the Chemical Reaction Engineering Module). Some of these solvers are connected to special functionality in the add-on modules.

- The Relationship Between Study Steps and Solver Configurations
- About the Stationary Solver
- The Eigenvalue Solver Algorithm
- About the Time-Dependent Solver

Remarks on Solver-Related Model Characteristics

THE IMPORTANCE OF A CORRECT JACOBIAN MATRIX

The solvers break down each problem—linear or nonlinear—into one or several linear systems of equations by approximating the given problem with a linearized problem. The coefficient matrix of the discretized linearized problem is called the *Jacobian matrix* (or *stiffness matrix*). In most cases COMSOL Multiphysics computes a correct Jacobian matrix.

The consequences of an incorrect Jacobian matrix depend on the solver in use:

- The linear stationary solver and the eigenvalue solver simply give an incorrect solution.
- The nonlinear stationary solver and the time-dependent solver take longer time to converge to the correct solution, and in some cases the solver even fails to find a solution. However, if the ignored terms have a very weak dependence on the sought solution, the impact on convergence speed is small.

An incorrect Jacobian matrix can occur in the following cases:

- If you supply an incorrect derivative of some of the user-defined functions and then use that function in some PDE coefficient or boundary condition specification with arguments that depend on the solution (for example, with the temperature as the function argument in a heat-transfer simulation).
- If you use a MATLAB M-file function or the External function interface, for which you have specified no derivative, and then call this function in some PDE or boundary condition with arguments that depend on the solution. A correct Jacobian can be computed if the function derivative is defined.
- If you use nonanalytic functions in a complex-valued problem, such as `real`, `imag`, `conj`, `abs`, or `arg`.
- If you use the `nojac` operator.

- About Global and Local Definitions
- MATLAB

WORKING WITH COMPLEX-VALUED PROBLEMS

When a problem contains complex numbers, be sure to consider the following two aspects:

- For time-dependent problems, the time-stepping algorithm must know if a problem is complex valued. If your model uses a complex-valued initial solution, COMSOL detects this and classifies the problem accordingly. If the problem is complex-valued because of a complex-valued PDE coefficient or other material property, go to the Time-Dependent Solver settings window, and in the **Advanced** section select the **Allow complex numbers** check box.
- If you expect to receive complex outputs from real inputs in elementary functions such as `sqrt`, `log`, and `asin`, add an Advanced subnode. Then in the settings window under **General**, select the **Allow complex-valued output from functions with real input** check box. By default, COMSOL gives an error message if a real input to a function generates a complex output.

Scaling of Variables and Equations

If the Dependent Variables in a model have widely different magnitudes, the solver might have problems with the resulting ill-conditioned matrix. The scaling of the dependent variables also influences the weighted errors computed by the solvers. For instance, in a structural mechanics problem the displacements can be of the order of 0.0001 m while the stresses are 1,000,000 Pa (1 MPa). To remedy this situation, COMSOL internally rescales the variables so that a well-scaled system results.

The rescaling of the discretized linear system occurs before constraint handling. Assume that the degrees of freedom U_i are expressed terms of rescaled degrees of freedom $\tilde{U}_i$ according to the formula

$$U_i = s_i\tilde{U}_i$$

where s_i are positive scale factors. Using a diagonal matrix S, the relation between U and $\tilde{U}$ is $U = S\tilde{U}$, and you can write the rescaled linear system as

$$\begin{bmatrix}\tilde{K} & \tilde{N}_F \\ \tilde{N} & 0\end{bmatrix}\begin{bmatrix}\tilde{U} \\ \tilde{\Lambda}\end{bmatrix} = \begin{bmatrix}\tilde{L} \\ \tilde{M}\end{bmatrix}$$

where

$$\Lambda = R\tilde{\Lambda} \qquad \tilde{N}_F = SN_FR \qquad \tilde{K} = SKS \qquad \tilde{N} = RNS$$

and

$$\tilde{L} = SL, \quad \tilde{M} = RM$$

Here, R is a diagonal matrix of positive scale factors chosen such that the rows in the matrix $\tilde{N}$ are of magnitude 1.

About the Stationary Solver

The following background information about the Stationary Solver discusses these topics—Damped Newton Methods, Linear Solvers vs. Nonlinear Solvers, Termination Criterion for the Fully Coupled and Segregated Attribute Nodes, and Pseudo Time Stepping. Also see Selecting a Stationary, Time-Dependent, or Eigenvalue Solver.

Stationary in the *COMSOL API Reference Manual*

DAMPED NEWTON METHODS

The nonlinear solver uses an affine invariant form of the damped Newton method as described in Ref. 3. You can write the discrete form of the equations as $f(U) = 0$, where $f(U)$ is the residual vector and U is the solution vector. Starting with the initial guess U_0, the software forms the linearized model using U_0 as the linearization point. It solves the discretized form of the linearized model $f'(U_0)\delta U = -f(U_0)$ for the Newton step δU using the selected linear system solver ($f'(U_0)$ is the Jacobian matrix). It then computes the new iteration $U_1 = U_0 + \lambda\delta U$, where λ ($0 \le \lambda \le 1$) is the damping factor. Next the modified Newton correction estimates the error E for the new iteration U_1 by solving $f'(U_0)E = -f(U_1)$. If the relative error corresponding to E is larger than the relative error in the previous iteration, the algorithm reduces the damping factor λ and recomputes U_1. This algorithm repeats the damping-factor reduction until the relative error is less than in the previous iteration or until the damping factor underflows the minimum damping factor. When it has taken a successful step U_1, the algorithm proceeds with the next Newton iteration.

A value of $\lambda = 1$ results in Newton's method, which converges quadratically if the initial guess U_0 is sufficiently close to a solution. In order to enlarge the domain of attraction, the solver chooses the damping factors judiciously. Nevertheless, the success of a nonlinear solver depends heavily on a carefully selected initial guess, so you should provide the best value for U_0, giving at least an order of magnitude guess for different solution components.

TERMINATION CRITERION FOR THE FULLY COUPLED AND SEGREGATED ATTRIBUTE NODES

You specify the termination criteria in the settings window for a Fully Coupled or Segregated subnode to the Stationary Solver node. Also see The Segregated Solver (Termination Criterion for a Segregated Solver).

Termination Criterion: Solution

For **Termination criterion: Solution**, the nonlinear iterations terminate when the following convergence criterion is satisfied: Let U be the current approximation to the true solution vector, and let E be the estimated error in this vector. The software stops the iterations when the relative tolerance exceeds the relative error computed as the weighted Euclidean norm

$$\text{err} = \sqrt{\frac{1}{M}}\sqrt{\sum_{j=1}^{M}\frac{1}{N_j}\sum_{i=1}^{N_j}\left(\frac{|E_{i,j}|}{W_{i,j}}\right)^2}$$

Here *M is the number of fields;* N_j is the number of degrees of freedom in field j. The double subscript denotes degree of freedom index (i) and field (j) component. We let $W_{i,j} = \max(|U_{i,j}|, S_j)$, where S_j is a scale factor that the solver determines from the scaling method. You select the scaling method from the **Method** list in the **Scaling** section of the Dependent Variables settings window. The solver then computes the scale factor S_j using the following rules:

- For **Automatic**, S_j is the average of $|U_{i,j}|$ for all DOFs i for fixed j, times a factor equal to 10^{-5} for highly nonlinear problems or 0.1 otherwise.
- For **Manual**, S_j is the value given in the **Scale** field.
- For **Initial value based**, S_j is the average of $|V_{i,j}|$ for all DOFs i with fixed j, where $V = U_0$ is the solution vector corresponding to the initial value. In case all DOFs are zero for that particular field j, the total mean of $|V_{i,j}|$ for all i and j is used instead.
- For **None**, $W_{i,j} = 1$. In this case, err is an estimate for the absolute error.

The (automatically damped Newton) nonlinear solver only checks the convergence criterion if the damping factor for the current iteration is equal to 1. Thus, the solver continues as long as the damping factor is not equal to 1 even if the estimated error is smaller than the requested relative tolerance.

Termination Criterion: Residual

For **Termination criterion: Residual**, the nonlinear iterations terminate when the following convergence criterion is satisfied: The software stops the iterations when the relative tolerance exceeds the relative error computed as the weighted Euclidean norm

$$\text{err} = \sqrt{\frac{1}{M}}\sqrt{\sum_{j=1}^{M}\frac{1}{N_j\tilde{W}_j^2}\sum_{i=1}^{N_j}|F_{i,j}|^2}$$

where F is the current residual and $\tilde{W}$ are the weights determined by the first and, if applicable, also the second residual. Here, the double subscript denotes the degree of freedom index (i) and the field (j) component. The iterations can also terminate if the relative step size is in the range of a hundred machine epsilon and in addition a full Newton step is taken.

Termination Criterion: Solution or Residual

For **Termination criterion: Solution or residual**, the nonlinear iterations terminate when the relative tolerance exceeds the relative error computed as the minimum of the solution based error and the error given by the **Residual factor** times the residual-based error above.

LINEAR SOLVERS VS. NONLINEAR SOLVERS

Automatic Nonlinearity Detection

COMSOL automatically detects nonlinearity, so you normally do not need to decide whether to use a linear or a nonlinear solver.

The automatic detection works through analysis of the variables contributing to the residual Jacobian matrix and the constraint Jacobian matrix. If the algorithm finds that both these matrices are complete and do not depend on the solution, the stationary solver (including parametric sweeps) uses a linear solver algorithm. Otherwise, the solver uses a nonlinear solver algorithm. "Complete" here means that the algorithm only found contributing variables for which the correct Jacobian is computed.

Overriding the Automatic Nonlinearity Detection

In some cases you might want to specify explicitly that the stationary solver uses the linear or nonlinear solver algorithm. Such cases include:

- Linear models where the automatic detection of linearity makes COMSOL use the nonlinear solver. This can happen, for example, for models that involve some less common types of coupling variables (directly or indirectly as part of some boundary conditions). The nonlinear solver usually converges directly for linear problems, but if that is not the case, you can switch to the linear solver.
- Using the linear solver to single-step Newton's method for a nonlinear problem.
- Using the linear solver to solve a linearized (nonlinear) problem.

Which Models are Nonlinear?

How do you determine if a problem is linear or nonlinear? Finding out is not always easy, but for most physics you can apply the following criterion: if any coefficient or material property contains a dependent variable the model is nonlinear. The same holds true for models based on a PDE in the coefficient form, again with the same criterion.

Note a few special cases that arise with some physics. First, in the Heat Transfer interfaces, if you include radiation terms for black-body radiation, which depend on temperature according to the Stefan-Boltzmann law, the problem is nonlinear. Second, Single-Phase Flow is always nonlinear unless $\rho = 0$, resulting in the linear Stokes equations.

Whether your problem is linear or nonlinear, the solvers break it down into one or several linear systems of equations. Therefore, the linear solver selection affects the solution time and memory requirements also for nonlinear models.

PSEUDO TIME STEPPING

Pseudo time stepping is used in transport problems to stabilize the convergence toward steady state. Here an adaptive feedback regulator controls a CFL (Courant–Friedrichs–Lewy) number which is then used for pseudo time stepping. The CFL number starts from a moderate value (order one) and increases up to several orders of magnitude at convergence.

A simple multiplicative PID regulator for CFL regulation is used

$$\mathrm{CFL}_{n+1} = \left(\frac{e_{n-1}}{e_n}\right)^{k_P}\left(\frac{\mathrm{tol}}{e_n}\right)^{k_I}\left(\frac{e_{n-1}/e_n}{e_{n-2}/e_{n-1}}\right)^{k_D}\mathrm{CFL}_n \qquad (19\text{-}1)$$

where the regulator parameters k_P, k_I, and k_D are positive constants. Here e_n is the nonlinear error estimate for step n and tol is a given target error estimate.

- The first factor is nothing but a power of the current convergence rate (based on the last two steps), and is the most important part of this regulator. If the error is decreasing, the regulator increases the CFL number and if the error is increasing, the regulator decreases the CFL number. The strength of this coupling (and the rapidness of this effect) is controlled by the parameter k_P.

- The next factor is used to regulate the CFL number toward the requested target error estimate. A standard local error estimate regulation uses only a factor of this sort, but for this type of regulation the absolute level of the error is not that important. However, without this factor (k_I=0) the CFL number might drift even though the error level is fluctuating on the same level. This factor can also be used to select an absolute regime for the error where increasing the CFL number should be more difficult.
- The last factor is a derivative factor; it is affected by the change of the convergence rate.

A hard lower limit $\mathrm{CFL}_n \geq 1$ is used, and to lower the risk of premature termination there is an extra requirement of not accepting convergence until $\mathrm{CFL}_n \geq \mathrm{CFL}_\infty = 10^4$.

Pseudo time stepping is available for stationary problems. In the coupled approach it functions together with the constant damped Newton solver. See the settings for Fully Coupled and Segregated for related parameters.

About the Parametric Solver

The parametric solver supports two algorithms, continuation and no continuation (*plain sweep*). To use continuation you need to both select the **Auxiliary Sweep** check box as well as select one of the parameters as the continuation parameter from the list under **Study Extensions** on the Stationary or Frequency Domain settings window. Continuation can only be used for one parameter, the others are run as a plain sweep outside the continuation sweep.

When you add a Stationary or Frequency Domain study, a *parametric continuation solver* is used to find the solution to a sequence of stationary PDE problems that arise when you vary some parameter of interest. This can be any parameter that defines an equation, boundary condition, material property, or similar property of the physics but not parameters that, for example, vary the geometry or mesh (for such a parameterization, use a Parametric Sweep). The parametric solver can also prove useful when it is difficult to get convergence in a nonlinear model. You can then introduce a parameter such that the solution is easy if the parameter is small. Then, to obtain the solution for the desired value of the parameter, slowly increase its value. This way, the nonlinear solver gets a good initial guess based on the solution for the previous parameter value.

About the Time-Dependent Solver

The following background information about the Time-Dependent Solver discusses these topics—The Implicit Time-Dependent Solver Algorithms and BDF vs. Generalized-α. Also see Selecting a Stationary, Time-Dependent, or Eigenvalue Solver.

🔍	Time in the *COMSOL API Reference Manual*

THE IMPLICIT TIME-DEPENDENT SOLVER ALGORITHMS

The finite element discretization of the time-dependent PDE problem is

$$\begin{aligned} 0 &= L(U, \dot{U}, \ddot{U}, t) - N_F(U, t)\Lambda \\ 0 &= M(U, t) \end{aligned}$$

which is often referred to as the *method of lines*. Before solving this system, the algorithm eliminates the Lagrange multipliers Λ. If the constraints $0 = M$ are linear and time independent and if the constraint force Jacobian N_F is constant then the algorithm also eliminates the constraints from the system. Otherwise it keeps the constraints, leading to a differential-algebraic system.

In COMSOL, the *IDA* and *generalized-a* solvers are available to solve the above ODE or DAE system:

- IDA was created at the Lawrence Livermore National Laboratory (Ref. 4) and is a modernized implementation of the DAE solver DASPK (Ref. 5), which uses variable-order variable-step-size backward differentiation formulas (BDF).
- Generalized-α is an implicit, second-order accurate method with a parameter α or ρ_∞ ($0 \le \rho_\infty \le 1$) to control the damping of high frequencies. With $\rho_\infty = 1$, the method has no numerical damping. For linear problems this corresponds to the midpoint rule. $\rho_\infty = 0$ gives the maximal numerical damping; for linear problems the highest frequency is then annihilated in one step. The method was first developed for second-order equations in structural mechanics (Ref. 7) and later extended to first-order systems (Ref. 8).

For implicit time-stepping schemes, a nonlinear solver is used to update the variables at each time step. The nonlinear solver used is controlled by the active Fully Coupled or Segregated solver subnodes. These subnodes provide much control of the nonlinear solution process: it is possible to choose the nonlinear tolerance, damping factor, how often the Jacobian is updated, and other settings such that the algorithm solves the nonlinear system more efficiently.

For the BDF (IDAS) solver there is another alternative available, and that is to use the nonlinear solver built-in IDAS. This solver is used when all the **Fully Coupled** and **Segregated** nodes are disabled. The linear solver is in this case is controlled by the active linear solver subnode.

The linearization of the above system used in the Newton iteration is

$$\begin{aligned} E\ddot{V} + D\dot{V} + KV &= L - N_F\Lambda \\ NV &= M \end{aligned}$$

where $K = -\partial L/\partial U$ is the stiffness matrix,

$$D = -\partial L/\partial\dot{U}$$

is the damping matrix, and

$$E = -\partial L/\partial\ddot{U}$$

is the mass matrix. When $E = 0$, D is often called the mass matrix.

When using IDA for problems with second-order time derivatives ($E \neq 0$), extra variables are internally introduced so that it is possible to form a first-order time-derivative system (this does not happen when using generalized-α because it can integrate second-order equations). The vector of extra variables, here U_v, comes with the extra equation

$$\dot{U} = U_v$$

where U denotes the vector of original variables. This procedure expands the original ODE or DAE system to double its original size, but the linearized system is reduced to the original size with the matrix $E + \sigma D + \sigma^2 K$, where σ is a scalar inversely proportional to the time step. By the added equation the original variable U is therefore always a differential variable (index-0). The error test excludes the variable U_v unless consistent initialization is on, in which case the differential U_v-variables are included in the error test and the error estimation strategy applies to the algebraic U_v-variables.

Absolute Tolerance Settings for the Time-Dependent Solver

For the Time-Dependent Solver under the section **Absolute Tolerance**, the absolute and relative tolerances control the error in each integration step. More specifically, let U be the solution vector corresponding to the solution at a certain time step, and let E be the solver's estimate of the (local) error in U committed during this time step. For the **Unscaled Method**, the step is accepted if

$$\left(\frac{1}{M}\sum_j \frac{1}{N_j}\sum_i \frac{|E_iU_i|}{A_{\mathrm{us},\,i}+R|U_i|}^{2}\right)^{1/2} < 1$$

where $A_{\mathrm{us},i}$ is the unscaled absolute tolerance for DOF i, R is the relative tolerance, M is the number of fields, and N_j is the number of degrees of freedom in field j. The numbers $A_{\mathrm{us},i}$ are computed from a conversion of the input value A_k for the corresponding dependent variable k. For degrees of freedom for Lagrange shape functions or for ODEs these values are the same as entered (that is $A_{\mathrm{us},\,i} = A_k$) but for vector elements there is a field-to-DOF conversion factor involved.

For the **Scaled Method** and when you select **Update scaled absolute tolerance**, the step is accepted if

$$\left(\frac{1}{M}\sum_j \frac{1}{N_j}\sum_i \left(\frac{|E_iY_i|}{A_{\mathrm{s},\,i}+R|Y_i|}\right)^{2}\right)^{1/2} < 1$$

where $A_{\mathrm{s},i}$ is the scaled absolute tolerance for DOF i, M is the number of fields, R is the relative tolerance, N_j is the number of degrees of freedom in field j, and Y_i is the scaled solution vector. For dependent variables that are using the scaling method **Automatic**, the numbers $A_{\mathrm{s},i}$ are computed from the input values A_k according to the formula

$$A_{\mathrm{s},\,i} = A_{k_i}(\beta + \|Y\|_{2,\,i}) \qquad \beta = \begin{cases} (1-e^{-\alpha j})\|Y\|_{\infty,\,k_i} + e^{-\alpha j} & 0 < \|Y\|_{\infty,\,k_i} < 1 \\ 1 & \text{else} \end{cases}$$

where $\alpha = \frac{1}{5}$, j is the time-step iteration number $j = 0,1,\ldots$, and $\|Y\|_{2,\,k_i}$, $\|Y\|_{\infty,\,k_i}$ are the 2-norm and maximum norm of the dependent variable k_i, respectively. Here A_{k_i} is the converted input value A_k for the field k and DOF i. For dependent variables that are using another scaling method or when the **Update scaled absolute tolerance** check box is cleared, then $A_{\mathrm{s},\,i} = A_{k_i}$.

If the solution is smaller than the absolute tolerance, there is no accuracy at all.

For DAEs (differential-algebraic equations), you can exclude the algebraic equations from the error estimation so that the error is only based on the differential equations. See Advanced for information about excluding algebraic equations from the error estimate.

BDF VS. GENERALIZED-α

The BDF solver uses backward differentiation formulas with order of accuracy varying from one (that is, backward Euler) to five. BDF methods have been used for a long time and are known for their stability. However, they can have severe damping effects, especially the lower-order methods. Backward Euler severely damps any high frequencies. Even if you are expecting a solution with sharp gradients, you might get a very smooth solution due to the damping in backward Euler.

The generalized-α (alpha) solver has properties similar to the second-order BDF solver but the underlying technology is different. It contains a parameter, called α in the literature, to control the degree of damping of high frequencies. Compared to BDF (with maximum order two), generalized-α causes much less damping and is thereby more accurate. For the same reason it is also less stable.

The implementation of the generalized-α method in COMSOL detects which variables are first order in time and which variables are second order in time and applies the correct formulas to the variables.

For most problems, generalized-α is an accurate method with good enough stability properties. Many physics interfaces in COMSOL—for transport problems, for example—have generalized-α as the default transient solver. Some complicated problems, however, need the extra robustness provided by the BDF method. There are also some problem types, like ODE systems, that can benefit from the higher accuracy that high-order BDF methods provide.

About the Time Discrete Solver

The Time Discrete Solver is used to solve fluid dynamics problems using a projection method. Also, you can use the time-discrete solver to solve problems that have been discretized in time using the `prev` or `bdf` operator.

In the time-discrete solver you cannot use the `d` operator to get the time derivatives. That is why the `prev` operator is needed so that time derivatives can be written using the backward Euler method. However, the `prev` operator does not make it possible to define variables like this:

a=f(prev(a))

That is, you cannot use the time-discrete solver to solve incremental problems in time because a variable cannot be expressed in terms of itself. Such a formulation leads to a circular variable dependency. If you want to implement a variable that is dependent on itself, that variable must be a dependent variable that you solve for as a continuous ODE or PDE.

TimeDiscrete in the *COMSOL API Reference Manual*

The Eigenvalue Solver Algorithm

The Eigenvalue Solver algorithm is described in this section. Also see Selecting a Stationary, Time-Dependent, or Eigenvalue Solver.

Finite element discretization leads to the generalized eigenvalue system

$$(\lambda-\lambda_0)^2 EU-(\lambda-\lambda_0)DU+KU+N_F\Lambda \;=\; 0$$

$$NU \;=\; 0$$

where the solver evaluates E, D, K, N, and N_F for the solution vector U_0; λ denotes the eigenvalue; and λ_0 is the linearization point. If $E = 0$, it is a linear eigenvalue problem; if E is nonzero, it is a quadratic eigenvalue problem. To solve the quadratic eigenvalue problem, COMSOL Multiphysics reformulates it as a linear eigenvalue problem. After constraint handling, it is possible to write the system in the form $Ax = \lambda Bx$.

More general eigenvalue problems sometimes arise when boundary conditions or material properties are nonlinear functions of the eigenvalue. These cases can be handled as a series of quadratic eigenvalue problems. COMSOL treats general dependences on the eigenvalue by assembling a quadratic approximation around the eigenvalue linearization point λ_0. Normally, iteratively updating the linearization point leads to rapid convergence.

Finding the eigenvalues closest to the *shift* σ is equivalent to computing the largest eigenvalues of the matrix $C = (A-\sigma B)^{-1}B$. To do this, the solver uses the ARPACK FORTRAN routines for large-scale eigenvalue problems (Ref. 9). This code is based on a variant of the Arnoldi algorithm called the *implicitly restarted Arnoldi method*

(IRAM). The ARPACK routines must perform several matrix-vector multiplications Cv, which they accomplish by solving the linear system $(A - \sigma B)x = Bv$ using one of the linear system solvers.

Eigenvalue in the *COMSOL API Reference Manual*

The Modal Solver Algorithm

The purpose of the Modal Solver is to speed up certain simulations by performing a model reduction using eigenpairs. That is, the solution of the underlying system of equations is assumed to be a linear combination of parametric or time-dependent coefficients and a few dominant eigenvectors.

The equation of interest can be written as

$$E\ddot{u} + D\dot{u} + Ku = L \tag{19-2}$$

where E is the mass matrix, D is the damping matrix, and K is the stiffness matrix. The vector L is, in general, time dependent. The algorithm assumes that the matrices are constant. Either E or D can be identically zero. When performing frequency response studies, the vector L is assumed to be harmonic. Before solving Equation 19-2, an eigenvalue analysis has to be performed. Let Φ_r be a matrix containing (right) eigenvectors from the eigenvalue analysis as columns. Let Φ_l be a matrix of the same size as Φ_r containing, for instance, the same vectors as Φ_r or left eigenvectors for the same setup as used to compute Φ_r. If all appearing Dirichlet boundary conditions are homogeneous, an approximation, u_m, of u can be written in terms of the eigenvectors:

$$u_m = \Phi_r q(t) \tag{19-3}$$

where $q(t)$ is a vector of unknown coefficients. Replacing u in Equation 19-2 by $\Phi_r q(t)$ and premultiplying by $\overline{\Phi_l}^T$ results in

$$\overline{\Phi_l}^T E\Phi_r\ddot{q} + \overline{\Phi_l}^T D\Phi_r\dot{q} + \overline{\Phi_l}^T K\Phi_r q = \overline{\Phi_l}^T L \tag{19-4}$$

The damping matrix D is present when performing the eigenvalue analysis (when it is nonzero). No matter if a nonzero D is present or not it is possible to add damping by giving damping ratios per mode. Denote the damping ratio for mode i ξ_i and let ω_i^2 be the corresponding eigenvalue. The damping is then added to

$$\overline{\Phi_l}^T D\Phi_r \text{ as } \mathrm{diag}(2\xi_i\omega_i(E_r)_{ii})$$

where E_r is the reduced mass matrix.

Modal Reduced Matrices—Time-Dependent Study

The Modal Solver (using a Time-Dependent Modal study) can export matrices and the right-hand side for use in further simulations. The reduced matrices are the stiffness matrix

$$K_r = \overline{\Phi_l}^T K\Phi_r$$

the mass matrix

$$E_r = \overline{\Phi_l}^T E\Phi_r$$

the damping matrix

$$D_r = \overline{\Phi_l}^T D\Phi_r$$

the damping ratio matrix

$$D_{\text{ratio}} = \operatorname{diag}(2\xi_i\omega_i(E_r)_{ii})$$

and the projection matrix Φ_r . For the transient problem the reduced right-hand sides are the load vector, zero solution

$$L_0 = \overline{\Phi_l}^T L|_{u = 0}$$

the load vector

$$L_r = \overline{\Phi_l}^T L$$

and the load vector, deviation

$$L_p = L_r - L_0$$

Two simulations are needed for a transient problem if $L_p \neq 0$: one for L_p and one for $f(t)L_0$, where $f(t)$ is a time-dependent function that controls the load factor. The solution is then given by superposition of the two solutions. Note that the vector L_p can be nonzero for numerical reasons. If you believe this to be the case you can reduce the output times to get a more exact value for L_p. In most cases the solution is accurately computed by $f(t)L_0$.

Frequency Response

For a frequency response study the load L is assumed to be of the type

$$L(\Omega, t) = \tilde{L}(\Omega)e^{i\Omega t} \tag{19-5}$$

where Ω is the frequency of the forcing function. The steady-state solution of Equation 19-4 is then of the form

$$q(t) = ce^{i\Omega t} \tag{19-6}$$

Use the expression of L from Equation 19-5 and the expression of q from Equation 19-6 in Equation 19-4 and expand around Ω_0 , where Ω_0 is the first frequency given, to get

$$\left[\left(\frac{\Omega - \Omega_0}{2\pi}\right)^2 \overline{\Phi_l}^T \tilde{E}\Big|_{\Omega_0} \Phi_r - \left(\frac{\Omega - \Omega_0}{2\pi}\right) \overline{\Phi_l}^T \tilde{D}\Big|_{\Omega_0} \Phi_r + \overline{\Phi_l}^T \tilde{K}\Big|_{\Omega_0} \Phi_r\right] c = \overline{\Phi_l}^T \tilde{L}(\Omega) \tag{19-7}$$

An approximative solution to the original problem is given by $u_m = \Phi_r c$. No matter if a nonzero $\tilde{D}$ is present or not it is possible to add damping by giving damping ratios per mode. Denote the damping ratio for mode i ξ_i and let ω_i^2 be the corresponding eigenvalue. The damping is then added to

$$\overline{\Phi_l}^T \tilde{D}\Big|_{\Omega_0} \Phi_r \text{ as } \operatorname{diag}\left(\frac{\xi_i\omega_i(E_r)_{ii}}{2\pi}\right)$$

where E_r is the reduced mass matrix.This assumes that no non-homogeneous Dirichlet boundary conditions are present.

The only type of parameter dependent Dirichlet boundary conditions that are supported are those that can be written as a scalar frequency dependent function times a constant vector (that is, the constraint vector M can be written as $M = m(\Omega)M_0$). For non-homogeneous Dirichlet boundary conditions a particular solution is needed. Therefore a particular solution is computed from the nonreduced equation

$$\begin{cases} Ku_p = 0 \\ Nu_p = M \end{cases} \tag{19-8}$$

The term

$$\overline{\Phi}_l^T\left(\left(\frac{\Omega-\Omega_0}{2\pi}\right)^2\tilde{E}\Big|_{\Omega_0}u_p-\left(\frac{\Omega-\Omega_0}{2\pi}\right)\tilde{D}\Big|_{\Omega_0}u_p\right)$$

is then added to the right side of Equation 19-7 to homogenize the original problem. Once an approximative solution, u_h, has been found to the homogeneous problem using modal analysis an approximative solution of the nonhomogeneous problem is given by

$$u_m = u_p + u_h$$

Modal Reduced Matrices—Frequency Response Study

The Modal Solver (using a Frequency Domain Modal study) can export matrices and the right-hand side for use in further simulations. The reduced matrices are the stiffness matrix

$$K_r = \overline{\Phi}_l^T\tilde{K}\Big|_{\Omega_0}\Phi_r$$

the mass matrix

$$E_r = \overline{\Phi}_l^T\tilde{E}\Big|_{\Omega_0}\Phi_r$$

the damping matrix

$$D_r = \overline{\Phi}_l^T\tilde{D}\Big|_{\Omega_0}\Phi_r$$

the damping ratio matrix

$$D_{\text{ratio}} = \operatorname{diag}\left(\frac{\xi_i\omega_i(E_r)_{ii}}{2\pi}\right)$$

and the projection matrix Φ_r .

For frequency response the load vector is the only vector needed. In that case

$$L_0 = L_r = \overline{\Phi}_l^T\tilde{L}(\Omega) \text{ and } L_p = 0$$

The right-hand side (rhs) is given by

$$\text{rhs} = f(\Omega)L_r$$

where *f* is a function of frequency that controls the load factor. The equation to solve is given by

$$A_r = \left(\frac{\Omega-\Omega_0}{2\pi}\right)^2E_r-\left(\frac{\Omega-\Omega_0}{2\pi}\right)D_r-D_{\text{ratio}}+K_r = \text{rhs}$$

Note that no nonhomogenous Dirichlet boundary conditions are assumed.

Modal in the *COMSOL API Reference Manual*

The Time Explicit Solver Algorithms

The Time Explicit Solver Rung-Kutta and Adams-Bashforth methods are discussed in this section. For the nodal discontinuous Galerkin method, it is natural and most efficient to use an explicit time-stepping method. Other situations where it can be advantageous is when using only particle tracing or wave problems together with so-called mass lumping.

TimeExplicit in the *COMSOL API Reference Manual*

RUNGE-KUTTA METHODS

Explicit classical Runge-Kutta methods of order 1–4 are supported. Runge-Kutta 4 is the default choice for the discontinuous Galerkin method.

ADAMS-BASHFORTH METHODS

The third-order Adams-Bashforth multistep method (AB3) for $u_t = R(u)$ is

$$u_{n+1} = u_n + \frac{k}{12}(23R(u_n) - 16R(u_{n-1}) + 5R(u_{n-2}))$$

where u_n is the solution at time t_n, and k is the time step.

The time restriction for the discontinuous Galerkin method for wave problems is directly proportional to the smallest mesh element size.

The Wave Form PDE Interface

For The Wave Form PDE Interface a suitable time step can be determined automatically: and specify the variable `wahw.wtc`, for example. In practice it is not uncommon that only a few elements are small, yet these elements dictate the overall time step for the problem. To remedy this problem, an Adams-Bashforth 3 (local) time-stepping scheme, which saves computational effort by time marching groups of elements of similar size independently, is available for the Wave Form PDE interface.

AWE Solver

Use the **AWE Solver** ( ) to perform fast-frequency parameter sweeps using asymptotic waveform evaluation (AWE). If, for a Frequency Domain study, the **Use asymptotic waveform evaluation** check box is selected under **Study Extensions**, this solver is used. It is an alternative way to perform parameter stepping to the one you get by using the Stationary Solver node in conjunction with the Parametric attribute subnode.

AWE in the *COMSOL API Reference Manual*

GENERAL

Use the **Parameter name** field to specify a parameter name. The use of several parameter names is not supported.

Use the **Parameter values** field to enter a vector of parameter values that define the parameter value span for the simulation. Exactly how the vector of parameter values is used by the solver is determined by the option **Parameters to store** in the **Output** section as described below.

An alternative to specifying parameter values directly in the **Parameters values** field is to specify them in a text file. You can use the **Load parameter values** field and the **Browse** button to specify such a text file. Use the **Read File** button to read the specified file. The read values appear in the **Parameters values** field.

!	Loading values from a file overwrites any values already present in this field. The format of the text files must be such that the parameter values appear one per row.

Use the **Expressions** field to specify a space-separated list of globally available scalar-valued expressions to be used for error estimation by the AWE algorithm.

TOLERANCES

In the AWE algorithm, the values of the expressions specified in the **Expressions** field in the **General** section are evaluated at one or more points of a parameter interval using certain expansions. The AWE algorithm is considered to have converged in that interval if the functional values resulting from the different expansions and evaluation points are similar enough. Use the:

- **Relative tolerance** field to specify to what relative tolerance the functional values must agree at the evaluation points.
- **Absolute tolerance** field to specify to what absolute tolerance the functional values must agree at the evaluation points.

EXPANSION SETTINGS

Use the **Evaluation points** field to specify a scalar or vector of values where the expressions defined by the **Expressions** field in the **General** section are to be evaluated. The evaluation points must be specified as a number between 0 and 1 because they are interpreted as being relative to the parameter interval under consideration. Entering a scalar value of 0.5 means that the expressions are evaluated at the midpoint of each interval. Use the:

- **Expansion size** list to specify the number of terms to include when performing Taylor expansions of the solution.
- **Expansion type** list to specify which expansion type to use when evaluating the solution at the different evaluation points:
  - Select **Padé** to compute Padé expansions based on the Taylor expansions. The Padé expansions are then used when evaluating the solution.
  - Select **Taylor** to use the Taylor expansion itself when evaluating the solution.

VALUES OF LINEARIZATION POINT

The problem solved by the AWE solver is assumed to be a linearization about a solution. You can specify such a solution (a linearization point) using the **Prescribed by** list. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point.

Use the **Solution** list to specify which solution to use if **Prescribed by** has been set to **Solution**. Select:

- **Zero** to use a linearization point that is identically equal to zero.
- Any other available solution to use it as linearization point. It can be the current solution in the sequence or a solution from another sequence or a solution that was stored with the Store Solution node. You select a stored solution by changing **Use** to the name of the stored solution.

To store the used linearization point in the output, select the **Store linearization point and deviation in output** check box.

OUTPUT

Use the **Parameters to store** list to control at what parameter values the solver stores a solution. Select:

- **Steps given** to store solutions at the parameter values entered in the **Parameter values** field in the **General** section.
- **Steps taken by solver** to store solutions at the parameter values where the AWE algorithm has performed an expansion.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk instead of in the computer's internal memory.

ADVANCED

By default the solver allows shorter intervals in the AWE algorithm than the relative tolerance (from the **Relative tolerance** field in the **Tolerances** section) times the length of the interval defined by the values in the **Parameter values** field in the **General section**. But if shorter intervals are detected, these intervals are not bisected and a warning is printed in the log. To modify the shortest allowed interval, select the **Minimal interval** check box and enter a limit for the interval length.

The **Accept short intervals** check box can be used to control how the solver handles intervals that are found to be too short. If this check box is cleared, the solver stops with an error if the interval found is too short. If you select the check box, the solver silently accepts short intervals.

Use the **Assembly strategy** list to control how the solver assembles quantities needed to compute a Taylor expansion. Select:

- **All** to assemble all quantities at once. This option is faster than **One**.
- **One** to assemble one quantity at a time. This option requires less memory than **All**.

Dependent Variables

The **Dependent Variables** node (u,v,w) handles initial data and scaling for the dependent variables that you solve for as well as how to compute dependent variables not solved for. The methods are applicable to the dependent variables present as Field subnodes ( ) under the Dependent Variables node. The Field node name matches the name of the variable.

The Dependent Variables node automatically updates the Field nodes. So, if the study type for the solver changes or if you use a different study type, then the Field nodes change accordingly. Also see Scaling of Variables and Equations for information.

To plot and evaluate the initial values for the dependent variables, right-click the **Dependent Variables** node and choose **Compute to Selected** ( ) or click the button the toolbar (this is similar to the **Get Initial Value** option for the main Study nodes).

The settings window has the following sections:

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to specify all settings locally in the **Dependent Variable** node's settings window. This setting relates to the Field node setting to **Solve for this field**.

INITIAL VALUES OF VARIABLES SOLVED FOR

This section is only available if you select **User defined** from the **Defined by study step** list.

Use the **Method** list to specify how to compute initial values for the dependent variables that you solve for. Select:

- **Initial expression** (the default) to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object. Use the **Solution** list to specify what solution object to use (directly or as part of the initial expression). Select:
  - **Zero** to initialize all variables to zero.
  - Any available solution object to use it as initial value.

Solution List Options by Study Type

Depending on the study type for the selected solution object, you can choose different solutions from a list underneath the **Solution** list:

- For a Stationary study, from the **Selection** list, select **Automatic** (the default) to use the last (typically the only) solution, select **First** to use the first (typically the only) solution, select **Last** to use the last (typically the only) solution, select **All** to use all (typically just one) solutions from that study, select **Manual** to use a specific solution number that you specify, or select **1** to use the first (typically the only) solution. If you use a parametric continuation of the stationary study, there can be additional solutions to choose from.
- For a Time Dependent study, from the **Time** list, select **Automatic** (the default) to use the solution for the last time, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Interpolated** to specify a time in the text field that opens and use the interpolated solution at that time, select **Manual** to use a specific solution number that you specify, or select one of the output times to use the solution at that time.
- For an Eigenvalue study, from the **Selection** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the eigenvalues to use the corresponding eigensolution.
- For a Parametric Sweep or Frequency Domain study, from the **Parameter value** list, select **Automatic** (the default) to use the first parameter value set or frequency, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the parameter value sets or frequencies to use the corresponding solution.

SCALING

Use the **Method** list to specify how to scale the variables solved for. Select:

- **Automatic** to get an automatically determined scaling (the default), which works well for most models. It is initially based on the magnitudes of the elements in the Jacobian and mass matrices. For nonlinear problems these scales are recomputed based on the magnitude of the solution iterate.
- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling if you know the order of magnitudes of the variables in advance. For instance, suppose that a model includes two fields `u` and `sigma`, and that the values of `u` are on the order of 10^{-4}, and the values of `sigma` are approximately 10^6. To use this knowledge, type `u 1e-4 sigma 1e6` in the **Scale** field that appears. The solvers then internally use a rescaled solution vector for the degrees of freedom for `u` and `sigma`, which both are of the order 1. The scaling factors are given per field (or per state for ODEs) and are internally

converted to degrees of freedom scaling factors. The units for the manual scales are the same as for the corresponding field quantity in the model’s base unit system.

- **None** to skip scaling.

The automatic scaling in COMSOL Multiphysics does not work when using the nonlinear stationary solver and a field or state has an identically zero solution (the solver does not converge). In this case use **Manual** or **None**.

The scaling method also can be specified per variable in the settings window for the variable’s **Field** node. For more information about scaling, see Scaling of Variables and Equations below.

VALUES OF VARIABLES NOT SOLVED FOR

This section is only available if you select **User defined** from the **Defined by study step** list.

These settings are only applicable if there are dependent variables in the model that you do not solve for (in the case, for example, when solving a multiphysics model using a sequential approach). Then use the **Method** list to specify how to compute the values of variables not solved for. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object. Use the **Solution** list to specify what solution object to use if **Method** has been set to **Solution**. Select:
  - **Zero** to initialize all variables to zero.
  - Any other available solution object to use it as initial value.

Depending on the solution object to use, you can choose different solutions to use. If a solution has nodes for storing solutions in its sequence you can choose which solution to use using the **Use** list. The **Current** value is the value that the solution has at the moment the value is read. The other values are the values stored in the respective nodes of the sequence.

See Solution List Options by Study Type for additional choices based on study type and then available in this section.

You select whether to solve for a variable or not by left-clicking a **Field** subnode (W U.T.P) and then select or clear the **Solve for this field** check box in the settings window.

OUTPUT

Use the **Keep solution** list to specify which dependent variables to make available for results analysis and visualization. This is only applicable when you evaluate the dependent variables by right-clicking the **Dependent Variables** node and choosing **Compute to Selected**. Select:

- **Initial values of variables solved for** (the default) to send the initial values of the variables solved for, as specified by the **Field** nodes, as the solution data for results analysis.
- **Values of variables not solved for** to send the values of the variables not solved for, as specified by the **Field** nodes, as the solution data for results analysis.

Eigenvalue Solver

Use the **Eigenvalue Solver** ( ) to find the solution to linear or linearized eigenvalue problems (also called eigenfrequency problems). This solver is automatically used when a Eigenvalue or Eigenfrequency study is added to the model.

Also see The Eigenvalue Solver Algorithm.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to specify the properties below (in addition to the relative tolerance, which is always available).

Use the **Desired number of eigenvalues** field to specify the number of eigenvalues and eigenvectors (default: 6) that the solver should compute and store in the output.

The number in the **Relative tolerance** field (default $1.0*10^{-6}$) controls the relative error in the computed eigenvalues.

From the **Eigenvalue transformation** list, select a transformation method for transforming the eigenvalues into another related quantity. The default is **None**, which keeps the original eigenvalues. Depending on the physics in the model other transformations might also be available.

Use the **Search for eigenvalues around** field to search for eigenvalues close to the specified real or complex scalar. This value is often called the shift. Select the **Transform value** check box to transform the search value using the selected eigenvalue transformation. The value of the eigenvalue is set in the **Value** field.

VALUES OF LINEARIZATION POINT

Both for linear and nonlinear PDE problems the eigenvalue problem is that of the linearization about a solution. Such a solution is specified with the **Prescribed by** list. Select:

- **Initial expression** (the default) to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point.

Use the **Solution** list to specify which solution to use if **Prescribed by** is set to **Solution**:

- Select **Zero** (the default) to use a linearization point that is identically equal to zero.
- Select any other available solution to use it as linearization point.

Select the **Store linearization point and deviation in output** check box to store the linearization point and the deviation from that linearization instead of the total solution.

If the eigenvalue itself appears nonlinearly, the solver reduces the problem to a quadratic approximation around an eigenvalue linearization point. Use the settings under **Value of eigenvalue linearization point** to specify such a scalar. Select the **Transform point** check box to transform the linearization point value using the selected eigenvalue transformation. Specify the value of the linearization point in the **Point** field (default value: 0).

OUTPUT

Select an option from the **Scaling of eigenvectors** list to specify the scaling method used to normalize the eigenvectors. Select:

- **RMS** to use root mean square normalization.

- **Max** to use maximum norm normalization.
- **Mass matrix** to scale the eigenvectors such that the modal masses become unity. If this scaling method is used the **Participation factor field** list displays. Select the field for which to compute the *mass participation factors* (typically a displacement field such as `comp1_u`).

Select the **Store solution out-of-core** check box to store the output solution on disk instead of in the computer's internal memory.

ADVANCED

The eigenvalue solver is an iterative algorithm. Use the **Maximum number of eigenvalue iterations** field to limit the number of iterations (default: 300).

Use the **Dimension of Krylov space** field to control the algorithm's memory use. The default value of 0 means that the solver sets the dimension automatically to approximately twice the number specified in the **Desired number of eigenvalues** field in the **General** section.

Use the **Eigenvalue search method around shift** list to control how the eigenvalue solver searches for eigenvalues around the specified shift value. Select:

- **Closest in absolute value** (the default) to search for eigenvalues that are closest to the shift value when measuring the distance as an absolute value.
- **Larger real part** to search for eigenvalues with a larger real part than the shift value.
- **Smaller real part** to search for eigenvalues with a smaller real part than the shift value.
- **Larger imaginary part** to search for eigenvalues with a larger imaginary part than the shift value.
- **Smaller imaginary part** to search for eigenvalues with a smaller imaginary part than the shift value.

LOG

The **Log** section contains logs of the eigenvalue solver results and properties of the assembled system, including the solver iterations and the total solution time. This log is stored in the Model MPH-file.

Conical Quantum Dot: model library path
COMSOL_Multiphysics/Equation-Based_Models/conical_quantum_dot

Modal Solver

Use the **Modal Solver** ( ) to perform either parameter stepping (also called frequency response) or time stepping (also called transient response) using a reduced model. The model reduction uses precomputed eigenvalues and eigenvectors. This solver is automatically used when a Time-Dependent Modal or Frequency Domain Modal study is added to the model.

Also see The Modal Solver Algorithm for more information.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step, or select **User defined** to specify all settings locally.

Use the **Study type** list to select the basic study type. Select:

- **Frequency domain** to perform parameter stepping using a reduced model. Then continue defining the settings For Frequency Domain Modal Studies.
- **Time dependent** to perform time stepping using a reduced model. Then continue defining the settings For Time Dependent Modal Studies.

For Frequency Domain Modal Studies

For a Frequency Domain Modal study, use the **Parameter values** field to enter a vector of parameter values that define the parameter value span for the frequency-domain simulation. Click the **Range** button ( ) to define a range of parameter values using the **Range** dialog box.

Exactly how the vector of parameter values is used by the solver is determined by the option **Parameter list type**.

An alternative to specifying parameter values directly in the **Parameters values** field is to specify them in a text file. You can use the **Load parameter values** field and the **Browse** button to specify such a text file. Click the **Read File** button to read the specified file. The read values appear in the **Parameters values** field.

> ! Reading parameter values from a file overwrites any values already present in that field. The format of the text files must be such that the parameter values appear one per row.

Use the **Parameter list type** list to control how to interpret the parameter values entered in the **Parameter values** field. Select:

- **Frequency** (the default setting) to use the parameter values without modification.
- **Fraction** to multiply the parameter values by the absolute value of the largest eigenvalue in the reduced model divided by two.
- **Spread** to treat the parameter values as an interval around each eigenvalue in the reduced model. That is, the absolute value of each eigenvalue is multiplied by the parameter values and the resulting parameter value vectors are concatenated into one.

Use the **Linearity** list to specify the type of linear behavior. Select:

- **Linear** to use a linear solver with the same linearization point for both residual and Jacobian computation, which corresponds to one step in Newton's method.
- **Linear perturbation** (the default setting) to use a linear solver that computes the Jacobian in the same way as the **Linear** option but uses a zero solution when computing the residual. It is useful for small-signal analysis and similar applications where the variations around a linearization point are of interest.

For Time Dependent Modal Studies

For a Time-Dependent Modal study, select a **Time unit** from the list. Then use the **Times** field to enter a vector of times that define the time span for the simulation. Click the **Range** button ( ) to define time values. Output from a simulation includes the times given in this field and the corresponding solutions.

Tolerance

Use the **Relative tolerance** field to enter a positive number (default value: 0.01). Depending on the selection in the **Study type** list in the **General** section, the tolerance means one of the following:

- When a Frequency Domain Modal study is selected, the **Relative tolerance** is used as a termination tolerance for iterative linear system solvers and for error checking (if enabled) for direct linear system solvers.
- When a Time-Dependent Modal study is selected, the **Relative tolerance** is used by the solver in each time step to control the relative error. The absolute tolerance settings below work in the same way as for the

time-dependent solver, but internally the full length absolute tolerance vector is transferred to the modes by the same transformation (projection) as is used to transform the problem to reduced form (the eigenmodes).

EIGENPAIRS

Use the **Solution** list to specify a solver configuration to be used when constructing the reduced model.

> The **Use** list is available for solution sequences with additional stored solutions. When available, select an option to specify a solution containing the modes to be used in the reduced model.

Use the **Eigenpairs** list to specify which of the eigenpairs present in the solution to include when constructing the reduced model. The default setting is **All** and the solver uses all available eigenpairs. Select **Manual** to enter a space-separated list of **Eigenpair numbers** in the field.

Use the **Damping ratios** field to enter either a scalar value or a space-separated list with values. The total number of entered values must be one or equal to the number of eigenpairs in the reduced model. If one number is entered, that value becomes the damping ratio for all eigenpairs. If the field is empty (the default), no damping is applied by the solver.

VALUES OF LINEARIZATION POINT

The problem solved by the **Modal Solver** is assumed to be a linearization about a solution. You can specify such a solution (a linearization point) with the **Prescribed by** list. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as a linearization point.
- **Solution** to use a solution as a linearization point. Then, when **Solution** is chosen from the **Prescribed by** list, specify which solution to use. Select:
    - **Zero** to use a linearization point that is identically equal to zero.
    - Any other available solution to use it as linearization point.

Select the **Store linearization point and deviation in output** check box to store the used linearization point in the output.

OUTPUT

The output from the solver can either be the solution, the reduced matrices, or both. Use the **Compute** list to specify **Solution**, **Solution and reduced matrices**, or **Reduced matrices**. If **Solution** or **Solution and reduced matrices** is selected, click to select the **Store solution out-of-core** check box to store the output solution on disk instead of in the computer's internal memory (this option is active by default).

If **Solution and reduced matrices** or **Reduced matrices** is selected, click to select one or more of the following check boxes—**Stiffness matrix**, **Damping matrix**, **Damping ratio matrix**, **Mass matrix**, **Projection matrix**, **Load vector**, **Load vector, zero solution**, or **Load vector, deviation**—to output the corresponding quantities.

ADVANCED

Use the **Load factor** field to enter a globally available scalar-valued expression (default: 1). The solver uses this expression to multiply the residual. The purpose is to facilitate the use of simple nonconstant Dirichlet boundary conditions (for frequency response) and simple nonconstant Neumann boundary conditions (for transient response).

Optimization Solver

The **Optimization Solver** ( ) provides the settings for solving PDE-constrained optimization problems. This solver requires the Optimization Module. See the *Optimization Module User's Guide* for details. Also see The Log Window (The Optimization Solver Log).

	Optimization in the *COMSOL API Reference Manual*

Plug Flow Solver

The **Plug Flow Solver** ( ) is the default solver for Stationary Plug Flow study steps, and it is a version of the Time-Dependent Solver, except it steps in *volume* instead of time. The correspondence between time, t, and volume, V, is given by $V = \nu t$, where ν is the volumetric flow rate (SI unit: m^3/s). It is specially designed to solve plug flow reactor models set up in the Reaction Engineering interface, which requires a license for the Chemical Reaction Engineering Module.

Stationary Solver

Use the **Stationary Solver** ( ) to find the solution to linear and nonlinear stationary problems (also called static or steady-state problems). This solver is automatically used when a Stationary or Frequency Domain study is added to the model.

Also see About the Stationary Solver for information about Damped Newton Methods, Linear Solvers vs. Nonlinear Solvers, and Pseudo Time Stepping.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

The number in the **Relative tolerance** field (default: 0.001) is used for tolerance-based termination of iterative solver processes and for error checking (if enabled) for direct linear system solvers.

	The termination tolerance used for iterative processes is also influenced by values specified in the **Tolerance factor** fields present in active Fully Coupled, Segregated, and Segregated Step subnodes. See Termination Criterion for the Fully Coupled and Segregated Attribute Nodes for details.

Use the **Linearity** list to specify whether to use a nonlinear or linear solver. Select:

- **Automatic** to perform an analysis that automatically detects if the problem can be solved with a linear solver approach. If this option is chosen, no other settings are required.
- **Linear** to use a linear solver. This option uses the same linearization point for both residual and Jacobian computation and corresponds to one step in Newton's method.
- **Linear perturbation** to use a linear solver. This option computes the Jacobian in the same way as the **Linear** option but uses a zero solution when computing the residual. It is useful for small-signal analysis and similar applications where the variations around a linearization point are of interest.
- **Nonlinear** to use a nonlinear solver. If this option is chosen, no other settings are required.

Values of Linearization Point

If **Linear** or **Linear perturbation** is selected, COMSOL Multiphysics assumes that the problem to be solved is a linearization about a solution. Specify such a solution (a linearization point) using the **Prescribed by** list**.** Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point. Use the **Solution** list to specify which solution to use if **Prescribed by** has been set to **Solution**. Select:
  - **Zero** to use a linearization point that is identically equal to zero.
  - Any other solution to use it as linearization point. It can be the current solution in the sequence or a solution from another sequence or a solution that was stored with the Store Solution node. You select a stored solution by changing **Use** to the name of the stored solution.

Select the **Store linearization point and deviation in output** check box to store the used linearization point. Also see Linear Solvers vs. Nonlinear Solvers.

OUTPUT

Select the **Reaction forces** check box to compute and store reaction forces in the output.

The computation of boundary flux variables involves solving a system of equations to obtain a continuous field from nodal flux values. If the **Use lumping when computing fluxes** check box is selected, this system of equations is lumped. The benefits of using this option is that it can avoid certain spurious oscillations in the computed flux field, and it is also slightly faster. Lumping is not suitable in 3D for shape functions of order higher than 1.

LOG

This section, which is initially empty, contains a log from the time stepping. It is not available when the stationary solver is a subnode to another solver. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is also available when reopening the model.

- Acoustics of a Muffler: model library path **COMSOL_Multiphysics/Acoustics/automotive_muffler**
- Deformation of a Feeder Clamp: model library path **COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp**

The Log Window

Time-Dependent Solver

Use the **Time-Dependent Solver** ( ) to find the solution to time-dependent problems (also called dynamic or unsteady problems) using the implicit time-stepping methods, BDF or generalized-α. This solver is automatically used when a Time Dependent study is added to the model.

Also see About the Time-Dependent Solver for information about The Implicit Time-Dependent Solver Algorithms and BDF vs. Generalized-α.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step (the default). If you select **User defined** (to override the settings defined in the corresponding study node) you can specify the following settings:

- Use the **Time unit** list to choose a time unit that is suitable for the time span of the simulation. The default time unit is inherited from the corresponding setting in the study step.
- Use the **Times** field to enter a vector of times that define the time span for the simulation using the **Range** button ( ) if needed (default: `range(0,0.1,1)`).
- Use the **Relative tolerance** field to enter a positive scalar number (default: 0.01). The solver uses this number to control the relative error in each time step.

ABSOLUTE TOLERANCE

See Absolute Tolerance Settings for the Time-Dependent Solver for details about this section.

Specify an absolute tolerance that is used by the solver to control the absolute error. The tolerance specified here is applied to all variables unless modified per variable by selecting another method than the global method for a variable.

Select a **Global method** to select how the specified absolute tolerance is to be interpreted for the variables that use the global method (by default, all variables use the global method). Select:

- **Scaled** to let the absolute tolerance be applied to scaled variables.
- **Unscaled** to let the absolute tolerance be applied to unscaled variables.

In the **Tolerance** field enter a positive number that is applied to either scaled or unscaled variables.

To specify the absolute tolerance individually for a variable, select from the **Variables** list and modify the corresponding tolerance using the **Method** list. Select:

- **Scaled** to apply the specified tolerance to scaled variables.
- **Unscaled** to apply the specified tolerance to unscaled variables.
- **Use global** (the default) to apply the tolerance specified for the global tolerance.

If **Scaled** or **Unscaled** is selected as the **Method**:

- Enter a **Tolerance** value to modify the absolute tolerance for the selected variable.
- If a problem of wave-equation type is being solved, *and* if **Method** in the **Time Stepping** section is set to **BDF**, then by default, the solver chooses a tolerance for these components. To manually enter a tolerance for a time derivative, select the **Tolerance, time derivative** check box and enter a tolerance in the associated field.

> The **Method** setting (**Scaled** or **Unscaled**) that is selected for a variable applies also to its time derivative.

Select the **Update scaled absolute tolerance** check box as required. See Absolute Tolerance Settings for the Time-Dependent Solver for details.

TIME STEPPING

Select a time-stepping **Method**. See BDF vs. Generalized-α for details. Select:

- **BDF** to use a backward differentiation formula.

- **Generalized alpha** to use the generalized-α method.
- **Initialization only** to compute consistent initial values only and then stop. If this option is selected, no other settings are required.

The following settings are available when **BDF** or **Generalized alpha** is selected above.

If a Fully Coupled or Segregated attribute node is attached to a **Time** node, the settings for the nonlinear systems solved by the time-stepping methods come from that node.

The time-stepping method **Generalized alpha** requires a Fully Coupled or Segregated attribute node.

The time-stepping method **BDF** can be used without a **Fully Coupled** or **Segregated** attribute node. In such a situation, the **BDF** method uses an internal automatic nonlinear solver.

Steps Taken by Solver

To modify how the time-stepping methods select the time steps, choose an option from the **Steps taken by solver** list. Select:

- **Free** to let the time-stepping method choose time steps freely. The times specified in the **Times** field in the **General** section are not considered when a time step is chosen.
- **Intermediate** to force the time-stepping method to take at least one step in each subinterval of the times specified in the **Times** field in the **General** section.
- **Strict** to force the time-stepping method to take steps that end at the times specified in the **Times** field in the **General** section. The solver takes additional steps in between these times if necessary.
- **Manual** to override the automatic choice of time step with a manual choice.

Manual is only available for **Generalized alpha** and overrides the local error estimation made in each time step.

Further options that apply to one or several combinations (as indicated at each bullet) of selections made in the **Method** and **Steps taken by solver** lists are:

If **Free**, **Intermediate**, or **Strict** is chosen for the **Steps taken by solver**:

- **Initial step**. By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.
- **Maximum step**. By default the solver chooses a maximum time step automatically. Select the **Maximum step** check box for manual specification of a maximum time step.

If **Manual** is chosen for the **Steps taken by solver**:

- **Time step**. Enter a manual time step specification as a scalar, a vector of times, or an expression containing global variables in the **Time step** field.

If **BDF** is chosen as the time stepping **Method**:

- **Allow complex numbers**. Select the **Allow complex numbers** check box to be able to solve problems that are not automatically determined to be complex valued in a correct way.
- **Maximum BDF order**. This setting controls the maximum allowed degree of the interpolating polynomial of the BDF method.

- **Minimum BDF order**. This setting can be used to prevent the solver from decreasing the order of the BDF method below 2
- **Event tolerance**. This setting can be used to set the event tolerance (default value: 0.01), which is used for root finding of event conditions when using implicit events; see Implicit Event.

This does not apply to the start-up phase of the simulation.

If **Generalized alpha** is chosen as the time stepping **Method**:

- **Time step increase delay** (available if **Free**, **Intermediate**, or **Strict** is chosen for the **Steps taken by solver**). Select this check box and enter a positive integer in the field to make the solver more restrictive when increasing the time step. Entering 0 results in the same behavior as clearing the check box.
- **Amplification for high frequency**. Enter a number between 0 and 1 to control how much damping of high frequencies the solver provides. A value close to 0 results in efficient damping, while a number close to 1 results in little damping.
- **Predictor**. Select **Linear** to use linear extrapolation of the present solution to construct the initial guess for the nonlinear system of equations to be solved at the next time step. Select **Constant** to use the current solution as initial guess.

RESULTS WHILE SOLVING

This section mirrors what is defined for the Time Dependent **Results While Solving** section. That is, changes made to the Time-Dependent node are reflected here.

OUTPUT

Use the **Times to store** list to control at what times the solver stores a solution. Select:

- **Specified values** to store solutions at the values entered in the **Times** field in the **General** section.
- **Steps taken by solver** to store solutions at the time steps taken by the solver.

The selection made in the list **Steps taken by solver** in the **Time Stepping** section influences the output in this situation.

- Select the **Store reaction forces** check box to compute and store reaction forces in the output.
- The computation of boundary flux variables involves solving a system of equations to obtain a continuous field from nodal flux values. If the **Use lumping when computing fluxes** check box is selected, this system of equations is lumped. The benefits of using this option is that it can avoid certain spurious oscillations in the computed flux field and it is also slightly faster. Lumping is not suitable in 3D for shape functions of order higher than 1.
- Select the **Store time derivatives** check box to store time derivatives of the variables solved for in the output. Storing the time derivatives gives more accurate results when evaluating quantities that involve these time derivatives.
- Select the **Store solution out-of-core** check box to store the output solution on disk rather than in the computer's internal memory.
- Select the **Store solution before and after events** check box to store two additional solutions every time an implicit or explicit event is triggered. See The Events Interface. This stores the solutions before and after the reinitialization.

ADVANCED

Use the **Singular mass matrix** list to control whether the solver automatically determines if a system includes a differential-algebraic equation or not. Select:

- **Maybe** to make the solver look for zero-filled rows or columns in the mass matrix as a means of detecting a differential-algebraic equation.
- **Yes** if the model includes a differential-algebraic equation where the mass matrix has no zero-filled rows or columns.

Use the **Consistent initialization** list to control how the solver performs consistent initialization of differential-algebraic systems. Select:

- **Backward Euler** to perform consistent initialization using a small artificial step with the backward Euler method. When this is selected, enter a value in the **Fraction of initial step for Backward Euler** field. This value is a dimensionless quantity that determines the size of the time step for the backward Euler method (in terms of the initial step). Adjusting this value can improve the accuracy of the initialization step but can also affect the start-up of some models. The default value is 0.001 (that is, the small backward Euler step size is 0.1% of the initial step size).
- **Off** to indicate that the initial values already are consistent, which means that the solver does not modify them.
- **On** to use a consistent initialization routine that is preferable to **Backward Euler** for index-1 differential-algebraic equations.

The **On** option is only available when **Time method** is set to **BDF** at the same time that the internal nonlinear solver of the BDF method is used.

Use the **Error estimation** list to control how to treat algebraic degrees of freedom of a differential-algebraic system when estimating the time discretization error. Select:

- **Include algebraic** (the default) to include the algebraic degrees of freedom in the error norm.
- **Exclude algebraic** to exclude the algebraic degrees of freedom from the error norm.

  Excluding algebraic degrees of freedom (which stem from stationary equations in the model) means that the algebraic variables are not included in the error test for the time step. The algebraic variables are still solved for as part of the general system of equations. Excluding the algebraic variables from the error test might have the effect that the constraints (including hidden constraints, which are implicitly part of the equations) are not accurately fulfilled. In general, excluding algebraic degrees of freedom is not recommended when solving DAE systems of index 1, whereas it can be generally encouraged for DAE systems of index 2 (see Ref. 6).

LOG

This section, which is initially empty, contains a log from the time stepping. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is available also when reopening the model.

The Log Window (The Time-Dependent Solver Log)

The Black-Scholes Equation: model library path
COMSOL_Multiphysics/Equation-Based_Models/black_scholes_put

Time Discrete Solver

Use the **Time Discrete Solver** ( ) to find the solution to time-dependent problems (dynamic or unsteady problems) that have already been discretized in time using, for example, the `prev` operator or the `bdf` operator. This solver is automatically used when a Time Discrete study is added to the model.

See About the Time Discrete Solver for background information.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Use the **Times** field to enter a vector of times that define the simulation's time span.

The time step is specified in the **Time step** field. Valid entries are a scalar, a vector of times, or an expression containing global expression variables.

Discretizing time derivatives using the `prev` operator or the `bdf` operator requires the solution at previous discrete times. How many previous time steps that should be stored is specified in the **Number of time discrete levels** field. If you, for example, use the first-order `bdf` operator (`bdf(u,1)`), the solution at one previous time step is required. Using the second-order `bdf` operator (`bdf(u,2)`) requires the solution at two previous time steps. The default value is 2.

Use the **Relative tolerance** field to enter a positive number. This number controls how accurately the nonlinear system of equations is solved in each time step. In general, the desired relative error in the solution should be entered here.

ABSOLUTE TOLERANCE

Here you can specify an absolute tolerance that the nonlinear solver uses to control the absolute error. The tolerance specified here is applied to all variables unless modified per variable by selecting another method than the global method for a variable.

Use the **Global method** list to select how the specified absolute tolerance is to be interpreted for the variables that use the global method (by default, all variables use the global method). Select:

- **Scaled** to let the absolute tolerance be applied to scaled variables.
- **Unscaled** to let the absolute tolerance be applied to unscaled variables.

In the **Tolerance** field you enter a positive number that is applied to either scaled or unscaled variables.

To specify the absolute tolerance individually for a variable, select the variable from the **Variables** list and modify the corresponding tolerance with the **Method** list. Select:

- **Scaled** to apply the specified tolerance to scaled variables.
- **Unscaled** to apply the specified tolerance to unscaled variables.
- **Use global** (the default) to apply the tolerance specified for the global tolerance.

If you select **Scaled** or **Unscaled** additional fields appear. Use the **Tolerance** field to modify the absolute tolerance for the selected variable.

RESULTS WHILE SOLVING

See Time-Dependent Solver for these settings.

OUTPUT

Use the **Times to store** list to control at what times the solver stores a solution. Select:

- **Specified times** to store solutions at the times entered in the **Times** field in the **General** section.
- **Steps taken by solver** to store solutions at the time steps taken by the solver.

When **Specified times** is selected, the solution to output is computed through interpolation. Therefore, the solution at previous time steps is not computed, which means that expressions with the `prev` and `bdf` operators cannot be used in analysis. Such expressions can only be used in analysis when you have selected **Steps taken by solver**.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk instead of in the computer's internal memory.

LOG

This section, which is initially empty, contains a log from the time stepping.

Time Explicit Solver

Use the **Time Explicit Solver** ( ) to find the solution to time-dependent problems (also called dynamic or unsteady problems) using the family of Runge-Kutta explicit time-stepping schemes or the Adams-Bashforth 3 solver. This solver is used with a Time Dependent study.

Also see The Time Explicit Solver Algorithms.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. You can also select **User defined** to define all settings locally.

Use the **Times** field to enter a vector of times that define the time span for the simulation. Click the **Range** button ( ) to define a range using the **Range** dialog box.

Use the **Method** list to specify the time-explicit method: **Adams-Bashforth 3**, **Adams-Bashforth 3 (local)** (available for the Wave Form PDE interface), or the classic **Runge-Kutta** family.

For Runge-Kutta, select the order of the time-stepping scheme from the **Order** list.

From the **Time stepping** list, for Runge-Kutta and Adams-Bashforth 3, specify **Manual** or time stepping **From expressions,** where the latter is useful for the Wave Form PDE. When you use **From expression**, a list of **Cell time scale expressions** appear, where you can add such expressions to define the time stepping. For explicit methods the largest stable time step can automatically be computed from an expression. Some physics interfaces (Wave Form PDE, for example) define such an expression in terms of an estimated maximum wave speed (defined by the interface) and the element size (`wahw.wtc`). Here the element order is also taken into account. The expression should in general represent a local cell time scale. For wave problems, the expression should be proportional to the time it takes for the fastest wave to pass one mesh element. Each expression given is evaluated on all mesh elements. The smallest value (time scale), over all elements and all expressions, dictates the time step used. If you select **User defined** from the **Defined by study step** list, you can use the **Add** button (+) and the **Delete** button ( ) to add or delete rows in the list.

The time step is specified in the **Time step** field when **Time stepping** manual is selected. Valid entries are a scalar, a vector of times, or an expression containing global expression variables. The default value is 0.001 s ($1e^{-3}$ s).

Use the **Linear solver** list to select the linear solver to be used within the time stepping scheme to invert the mass matrix. Available linear solvers appear in the model tree. The default is to use the **Direct** linear solver. For cheap but approximate inversion of the mass matrix use the **Lumped** option.

This option can only be used together with a linear space discretization.

In rare cases, when the PDE is nonlinear, you can adjust the **Relative tolerance** (default value: 0.01).

RESULTS WHILE SOLVING

See Time Dependent for these settings.

OUTPUT

See Time Discrete Solver for these settings.

References for the Solution Operation Nodes and Solvers

1. P.E. Gill, W. Murray, and M.A. Saunders, *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Systems Optimization Laboratory (SOL), Stanford University, 2006.

2. P.E. Gill, W. Murray, and M.A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

3. P. Deuflhard, "A Modified Newton Method for the Solution of Ill-conditioned Systems of Nonlinear Equations with Application to Multiple Shooting," *Numer. Math.*, vol. 22, pp. 289–315, 1974.

4. A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward, "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," *ACM T. Math. Software*, vol. 31, p. 363, 2005.

5. P.N. Brown, A.C. Hindmarsh, and L.R. Petzold, "Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems," *SIAM J. Sci. Comput.*, vol. 15, pp. 1467–1488, 1994.

6. K.E. Brenan, S.L. Campbell, and L.R. Petzold, *Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations*, pp. 146–147, SIAM, Philadelphia, Pa, 1996.

7. J. Chung, G.M. Hulbert, "A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized-α Method," *J. Appl. Mech.*, vol. 60, pp. 371–375, 1993.

8. K.E. Jansen, C.H. Whiting, G.M. Hulbert, "A Generalized-α Method for Integrating the Filtered Navier–Stokes Equations with a Stabilized Finite Element Method," *Comput. Methods Appl. Mech. Engrg.*, vol. 190, pp. 305–319, 2000.

9. The ARPACK Arnoldi package, www.caam.rice.edu/software/ARPACK.

10. P. Deuflhard, "A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques," *Numer. Math.*, vol. 33, pp. 115–146, 1979.

11. R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Teubner Verlag and J. Wiley, Stuttgart, 1996.

12. R. Rannacher, "A Feed-back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples," *East-West J. Numer. Math.*, vol. 4, pp. 237–264, 1996.

13. www.netlib.org/ode.

Solution Attribute Nodes

The first few sections provide some background information about the linear system solvers and preconditioners and the algorithms used:

- About the Advanced Attribute Settings
- Choosing the Right Linear System Solver
- About Incomplete LU
- The Adaptive Mesh Refinement Solver
- The Domain Decomposition Solver
- The Fully Coupled Attribute and the Double Dogleg Method
- The Iterative Solvers
- The Multigrid Solvers
- The Parametric Solver Algorithm
- The SCGS Solver
- The Segregated Solver
- The Sensitivity Analysis Algorithm
- About the SOR, SOR Gauge, SOR Line, and SOR Vector Iterative Solver Algorithms
- The Vanka Algorithm

About Solver Commands in the *COMSOL API Reference Manual*

Then the settings for the solver attribute nodes—such as preconditioners, adaptive mesh refinement, and sensitivity analysis listed in Table 19-6—are detailed. There is also a list of the References for the Linear System Solvers and the Preconditions.

TABLE 19-6: SOLUTION ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
	Adaptive Mesh Refinement	Handles adaptive mesh refinement together with a time-dependent solver. The adaptive mesh refinement then creates multiple meshes for segments of the time-dependent simulation. Add it together with Eigenvalue, Stationary, and Time-Dependent solvers. Also see The Adaptive Mesh Refinement Solver.
	Advanced	Advanced general solver parameters.
	AMS	Handles parameters for linear system solvers/preconditioners that use the auxiliary space Maxwell solver (AMS). Add it to Iterative, Krylov Preconditioner or Coarse Solver attributes.
	Automatic Remeshing	Adds automatic remeshing parameters. The remeshing occurs when the mesh quality falls below a specified value. Add it to a Time-Dependent solver.
	Coarse Solver	Handles settings for the coarse solver when using a Multigrid or Domain Decomposition.

TABLE 19-6: SOLUTION ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
⟳	Control Field	Handles settings for field variables that are acting as control variables. Control variables have a special status when using the Sensitivity or the Optimization solver. Used together with the Dependent Variables operation node.
⟳	Control State	Handles settings for state variables that are acting as control variables. Control (state) variables have a special status when using the Sensitivity or the Optimization solver. Used together with the Dependent Variables operation node.
▧	Direct	Handles settings for a direct linear solver.
⊂⊐	Domain Decomposition	Used to set up an additive-, multiplicative-, hybrid- or symmetric Schwartz overlapping domain decomposition solver. Add it to Iterative, Krylov Preconditioner or Coarse Solver attributes. Also see The Domain Decomposition Solver.
⟳	Domain Solver	Handles settings for the domain solver (when using Domain Decomposition).
W U.T.P	Field	Handles settings for field variables. Each field variable needs a separate Field node. This attribute is used with the Dependent Variables operation node.
✚	Fully Coupled	Uses a damped version of Newton’s method or a double dogleg method that handles parameters for a fully coupled solution approach. It can be used with the Stationary or Time-Dependent solvers. Also see The Fully Coupled Attribute and the Double Dogleg Method.
LU	Incomplete LU	Handles parameters for linear system solvers/preconditioners that use incomplete LU factorization. Add it to Iterative, Krylov Preconditioner or Coarse Solver attributes.
⧄	Iterative	Handles settings for an iterative linear solver or preconditioner. Also see The Iterative Solvers.
J	Jacobi	Handles settings for the Jacobi (or diagonal scaling) method. Add it to Iterative, Krylov Preconditioner, Presmoother, Postsmoother, or Coarse Solver attributes.
K	Krylov Preconditioner	Handles settings for a Krylov-type linear solver or preconditioner. Add it to Iterative, Presmoother, Postsmoother, or Coarse Solver attributes.
±	Lower Limit	Parameters for imposing restrictions on degrees of freedom. Add it to a Segregated attribute.
⊞	Lumped Step	Available with a Segregated attribute node. This step is intended for speeding up the computation of any L_2-projections, stemming from the identity operator, appearing as single physics within a multiphysics problem.
▦	Multigrid	Handles settings for a multigrid linear solver or preconditioner. Add it to Iterative, Krylov Preconditioner, Presmoother, Postsmoother, or Coarse Solver attributes. Also see The Multigrid Solvers.

TABLE 19-6: SOLUTION ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
	Parametric	Handles settings for parameter stepping. This attribute can be used together with the Stationary Solver. Also see The Parametric Solver Algorithm.
	Postsmoother	Handles settings for the postsmoother (when using Multigrid).
	Presmoother	Handles settings for the presmoother (when using Multigrid).
	Previous Solution	An optional attribute node of the Parametric attribute node. It handles field variables that have to be accessed at a previous parameter value or time.
S	SCGS	Handles the SCGS (symmetrically coupled Gauss-Seidel) solver, which is useful as a preconditioner for solving the Navier-Stokes equations and similar fluid-flow problems. Also see The SCGS Solver.
	Segregated	Handles parameters for a segregated solution approach. This attribute makes it possible to split the solution process into substeps. Each substep uses a damped version of Newton's method. Add it to Stationary and Time Dependent solvers. Also see The Segregated Solver.
	Segregated Step	Handles settings for one substep of a segregated iteration. This attribute uses a damped version of Newton's method and can be used together with a Segregated attribute node.
	Sensitivity	Sensitivity parameters. Also see The Sensitivity Analysis Algorithm.
SOR	SOR	Handles settings for the SOR (successive over-relaxation) iterative method. Add it to Iterative, Krylov Preconditioner, Presmoother, Postsmoother, or Coarse Solver attributes. Also see The SOR Method.
SOR	SOR Gauge	Handles settings for an SOR Gauge-type linear solver or preconditioner. Also see The SSOR Gauge, SOR Gauge, and SORU Gauge Algorithms.
SOR	SOR Line	Handles settings for an SOR Line linear solver or preconditioner. Also see The SOR Line Algorithm.
SOR	SOR Vector	Handles settings for an SOR Vector-type linear solver or preconditioner. Also see The SOR Vector Algorithm.
	State	Handles settings for state variables. A state is composed of a set of ODE variables. Used together with the Dependent Variables operation node.
	Stationary Acceleration	Accelerates the solution process for nonlinear problems with a time-periodic stationary solution.
STOP	Stop Condition	An attribute that stops parameter stepping or time stepping when a specified condition is fulfilled.
	Time Parametric	Handles settings for parameter stepping. This attribute can be used together with the Time-Dependent Solver.
V	Vanka	Handles settings for a Vanka linear solver or preconditioner. Also see The Vanka Algorithm.

About the Advanced Attribute Settings

Advanced in the *COMSOL API Reference Manual*

WHICH PROBLEMS ARE SYMMETRIC?

When using an Advanced attribute node, you have an option to choose the matrix symmetry. But how do you know which problems are symmetric? When the discretization of a PDE problem results in a symmetric Jacobian (stiffness) matrix (and a symmetric mass matrix for time-dependent or eigenvalue problems), you can often apply faster and less memory-consuming algorithms to solve the resulting linear systems. PDEs with symmetric discretization typically occur in models involving acoustics, diffusion, electromagnetics, heat transfer by conduction, and structural mechanics. In contrast, problems in fluid mechanics, convection-diffusion, and convection-conduction typically involve nonsymmetric Jacobian matrices.

If the model involves complex numbers you can distinguish between *symmetric* and Hermitian matrices. A *Hermitian matrix* A satisfies

$$\bar{A}^T = A$$

where T denotes the transpose and the bar denotes the complex conjugate.

COMSOL Multiphysics detects symmetry for symmetric and Hermitian matrices. To take advantage of the computational savings for models with symmetric matrices is to use a solver that utilizes the symmetry. The following linear system solvers and preconditioners do not take advantage of symmetric matrices:

- The Vanka preconditioner
- The incomplete LU preconditioner

The algebraic multigrid solver/preconditioner

Selecting **Symmetric** for a problem that does not result in symmetric matrices leads to an incorrect solution.

ELIMINATION CONSTRAINT HANDLING

The constraint handling is, for simplicity, demonstrated for a stationary problem. The handling is similar for parametric, eigenvalue, and time-dependent problems. Consider the linear (scaled) algebraic system

$$\begin{bmatrix} K & N_F \\ N & 0 \end{bmatrix} \begin{bmatrix} U \\ \Lambda \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix}$$

The Lagrange multiplier vector Λ is typically undetermined, and COMSOL Multiphysics does not solve for it. Similarly, the constraint $NU = M$ often contains the same equation several times. To handle this problem, COMSOL turns to a constraint-handling method that uses elimination. The solver computes a solution U_d to the constraint $NU = M$ as well as a matrix Null, whose columns form a basis for the null space of N. For unidirectional constraints ($N_F \neq N^T$) a matrix Nullf is also computed, whose columns form a basis for the null space of $N_\mathrm{F}{}^T$. Then it obtains the solution as $U = \mathrm{Null}\,U_n + U_d$. Here U_n is the solution of $K_c\,U_n = L_c$, where

$$\begin{cases} K_c = \mathrm{Nullf}^T K\,\mathrm{Null} \\ L_c = \mathrm{Nullf}^T (L - KU_d) \end{cases}$$

Here K_c is the eliminated stiffness matrix.

For eigenvalue and time-dependent problems, the corresponding eliminated D and E matrices are

$$D_c = \mathrm{Nullf}^T D\,\mathrm{Null}, \qquad E_c = \mathrm{Nullf}^T E\,\mathrm{Null}$$

Choosing the Right Linear System Solver

The following pertains to the Direct attribute node. All linear system solvers above work on general sparse linear systems of the form $Ax = b$ and use LU factorization on the matrix A to compute the solution x. In doing so, they use a preordering algorithm that permutes the columns of A to minimize the number of nonzeros in the L and U factors. Popular preordering algorithms include Minimum degree, Nested dissection, and Multisection. The MUMPS and SPOOLES solvers run distributed when running COMSOL Multiphysics in distributed mode (on clusters, for example). All linear system solvers benefit from shared memory parallelism (multicore processors, for example); however, MUMPS do so to a slightly lesser extent than PARDISO and SPOOLES.

This section reviews Linear System Solver Selection Guidelines, Which Models Are Positive Definite?, and Elliptic and Parabolic Models.

Linear in the *COMSOL API Reference Manual*

The MUMPS Solver

The MUMPS solver works on general systems of the form $Ax = b$ and uses several preordering algorithms to permute the columns and thereby minimize the fill-in. MUMPS is multithreaded on platforms that support multithreading and also supports solving on distributed memory architectures through the use of MPI. The code is written in Fortran 90. For further details about MUMPS, see Ref. 1.

The PARDISO Solver

The PARDISO solver works on general systems of the form $Ax = b$. In order to improve sequential and parallel sparse numerical factorization performance, the solver algorithms are based on a Level-3 BLAS update, and they exploit pipelining parallelism with a combination of left-looking and right-looking supernode techniques. PARDISO is multithreaded on platforms that support multithreading. On distributed memory architectures the solver settings are changed to corresponding MUMPS settings if needed. The code is written in C and Fortran. COMSOL uses the PARDISO version developed by Olaf Schenk and collaborators (Ref. 3), which is included with Intel MKL (Intel Math Kernel Libraries).

The SPOOLES Solver

The SPOOLES solver works on general systems of the form $Ax = b$ using the multifrontal method and direct LU factorization of the sparse matrix A. When the matrix A is symmetric or Hermitian, the solver uses an LDLT version of the algorithm, which saves half the memory. SPOOLES uses several preordering algorithms to permute the columns and thereby minimize fill-in. SPOOLES is multithreaded on platforms that support multithreading and also supports solving on distributed memory architectures through the use of MPI. The code is written in C. COMSOL uses SPOOLES version 2.2 developed by Cleve Ashcraft and collaborators (Ref. 2).

The Dense Matrix Solver

The dense matrix solver works on general systems of the form $Ax = b$. The dense matrix solver uses LAPACK (Ref. 4) for multithreaded solves and ScaLAPACK (Ref. 5) for distributed memory architectures. This solver is mainly useful for cases where the system matrices are densely populated, such as boundary element (BEM) models.

LINEAR SYSTEM SOLVER SELECTION GUIDELINES

The physics in the model selects a default linear system solver that usually is appropriate for the problem type, at least for single-physics models. If the default solver does not perform well, use the following guidelines to choose a linear system solver.

1. Try the PARDISO direct solver.
2. Try the MUMPS direct solver.
3. If the solver still runs out of memory or is too slow, use one of the iterative solvers GMRES, FGMRES, or BiCGStab. Select a preconditioner according to the guidelines in the section about the iterative solver.
4. If the system is positive definite and real symmetric or Hermitian, try the conjugate gradients iterative solver, which is more memory-efficient and sometimes faster than GMRES, FGMRES, and BiCGStab. Select a symmetric preconditioner. Alternatively, try the SPOOLES direct solver. It often uses less memory but is less numerically stable. SPOOLES is also slower.

WHICH MODELS ARE POSITIVE DEFINITE?

A model with a real symmetric or Hermitian system matrix is often also *positive definite*, which means that a number of efficient linear system solvers are applicable. Further, the simple preconditioners SSOR, SOR, SORU, Jacobi (diagonal scaling), and the multigrid solvers benefit from a positive definite matrix. A real symmetric or Hermitian matrix is positive definite if all its eigenvalues are positive.

For stationary problems, the system matrix is the Jacobian (stiffness) matrix A. This means that stationary models in diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix.

For time-dependent problems, the system matrix is of the form $A + \sigma B + \sigma^2 C$, where B is the damping matrix, C is the mass matrix, and $\sigma > 0$ is inversely proportional to the time step (if $C = 0$, then B is often called the mass matrix). Because these matrices are often positive definite, time-dependent models in diffusion, electromagnetics, structural mechanics, and heat transfer by conduction usually have a positive definite system matrix.

For eigenvalue problems, the system matrix is of the form $A - \sigma B + \sigma^2 C$, where σ is the *shift*—that is, the number around which the software searches for eigenvalues (specified in the **Search for eigenvalues around** field; the default is 0). Because A, B, and C are usually positive definite, eigenvalue problems in acoustics, diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix if $\sigma \le 0$.

ELLIPTIC AND PARABOLIC MODELS

The classes of elliptic and parabolic models include the positive definite models. For such models, the efficient multigrid preconditioners often perform well. A simplified definition of these classes reads as follows. A system of stationary or eigenvalue second-order PDEs is *elliptic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix. A system of time-dependent PDEs has a time derivative term of the form $d_a\dot{u}$, where the mass coefficient d_a is often a positive definite matrix and the e_a coefficient is 0. Such a system is *parabolic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix.

Stationary or eigenvalue models in acoustics, convection-diffusion, electromagnetics, heat transfer, and structural mechanics are usually elliptic. Likewise, time-dependent models in convection-diffusion, electromagnetics, and heat transfer are often parabolic. The Navier-Stokes equations, wave-type equations, or formulations involving weak constraints are neither elliptic nor parabolic.

About Incomplete LU

The Incomplete LU preconditioner performs an incomplete LU factorization of the system matrix A. That is, it drops small elements during the column-oriented Gaussian elimination (see Ref. 15 and Ref. 16). Thus it saves memory, and the resulting factors L and U are approximate. The resulting preconditioner is an approximation to A. The preconditioner supports threshold drop, fill-ratio drop, and threshold pivoting. It can optionally respect the nonzero pattern in the original matrix. The preconditioner accepts matrices in symmetric and Hermitian format but expands these to full storage before factorization.

SELECTING A DROP RULE

The Incomplete LU preconditioner uses the *threshold drop rule* (the default) or the *fill-ratio drop rule*. The preconditioner drops (neglects) an element during the elimination phase if its absolute value is smaller than the Euclidean norm of the entire column times a *drop tolerance*. In contrast, the fill-ratio drop rule limits the number of nonzeros in the incomplete factors L and U, and it keeps the largest absolute values. The number of values it keeps depends on the number of nonzeros in the corresponding column of the original matrix times a fill-ratio factor. There are two exceptions to these drop rules:

- The preconditioner never drops diagonal elements.
- The preconditioner optionally drops nonzeros in positions where the original matrix is nonzero. To make the preconditioner drop them, clear the **Respect pattern** check box in the settings for the Incomplete LU preconditioner.

The primary problem with setting up a preconditioner is the trade-off between resources (computer time and memory) and the preconditioner's quality. The computational cost of setting up a preconditioner with the Incomplete LU preconditioner is at least proportional to the number of nonzeros in the produced factors L and U. An advantage of using the fill-ratio drop rule is that you can estimate and limit the cost beforehand; the main disadvantage is that the quality of the preconditioner is typically not as good as using the threshold drop rule with a drop tolerance resulting in the same number of nonzeros. However, with the threshold drop rule there is no good way of estimating resource requirements beforehand. Furthermore, there is no general formula for these drop rules that gives a drop tolerance or fill ratio that guarantees fast convergence for a certain iterative method. Therefore it is often necessary to rely on experiments and experience for this difficult and, from a performance point of view, important choice.

The Adaptive Mesh Refinement Solver

The Adaptive Mesh Refinement solver algorithm and Error Estimates for the Time-Dependent Solver, The L_2 Norm Error Estimate, and The Functional Error Estimate are discussed in this section.

Adaption in the *COMSOL API Reference Manual*

THE ADAPTIVE SOLVER ALGORITHM

The adaptive solver performs the following iterative algorithm (Ref. 11):

1. Solve the problem on the existing mesh using the stationary or eigenvalue solver.
2. Evaluate the residual of the PDE on all mesh elements.
3. Estimate the error in the solution on all mesh elements. The computed error estimate is really an *error indicator* because the estimate involves an unknown constant (C above).

4 Terminate execution if it has made the requested number of refinements or if it has exceeded the maximum number of elements.

5 Refine a subset of the elements based on the sizes of the local error indicators.

6 Repeat these steps.

Time-dependent Adaption

An adapted solution at $t = t_n$ is mapped to the coarse base mesh. A new adapted mesh for the time interval $[t_n, t_{n+1}]$ is constructed by first computing a coarse solution on the base mesh in $[t_n, t_s]$, where t_s is the largest sample time and $t_{n+1} = 2t_n - t_{n-1}$. The error indicator is evaluated using the coarse solution at the given sample points.

In the case of an automatic time interval, a measure of the amount of refinement is computed and compared to a given requested value. If the computed value is too small or too large the interval length is increased or decreased, respectively, which results in a new t_{n+1}. If the interval length need to be changed the error indicator is sampled again using a new coarse solution. The comparison is done only once.

The new adapted mesh is obtained by using the error indicator sampled at given points in $[t_n, t_{n+1}]$, selecting a set of elements based on the element pick function, and then finally refining these elements. The solution at t_n to the PDE problem on the previous adapted mesh for $[t_{n-1}, t_n]$ is then mapped to the new mesh for $[t_n, t_{n+1}]$ and time integration continues until the next mesh adaption takes place att_{n+1}.

The simple measure used for determining the amount of refinement is

$$\rho = \frac{1}{2^p N} \sum_{i = 1, \gamma(i) \neq 0}^{N} 2^{\gamma(i)}$$

Here γ is an N-vector of integers containing the number of times the element at that position should be refined, $p = \max_i \gamma(i)$, and N is the number of elements of the coarse base mesh.

- Now, the next interval length is decreased by a given factor if ρ is larger than 120% of the requested reference value. If ρ is smaller than 80% of the reference value it is instead increased. Otherwise the time interval length is kept the same.

Adaptive mesh refinement works with the mesh and equations defined in the domains (interior) of the geometry and does not consider meshes and equations on lower dimensions, such as surface meshes in shell models. In solid geometries, the adaptive mesh refinement of the interior mesh usually affects the surface mesh.

ERROR ESTIMATES

Error Estimates for the Time-Dependent Solver

A possible error indicator is the L_2 norm of the gradient of the dependent variables (for example, `sqrt(comp1.Tx^2+comp1.Ty^2)` for the temperature in a 2D heat transfer model). The gradient of the dependent variable is the default value for the error indicator in most physics.

A solution on the coarse base mesh is computed in the next time interval, and the error indicator is evaluated at the points specified in the **Sample points** field. In this way a new adapted mesh appropriate for the next time interval can be generated. The sample points must be specified as a number between 0 and 1 because they are interpreted as being relative to the time interval under consideration. Entering a scalar value of 0.5 means that the error indicator is evaluated at the midpoint of the interval. The default value is `range(0.0,0.1,1.0)`, which gives 11 sample points from 0 to 1.

The L_2 Norm Error Estimate

The L_2 norm error estimate relies on an assumption of a strong stability estimate for the PDE problem (satisfied, for example, for Poisson's equation over a domain with a smooth boundary). From such an assumption, it is possible to show that there is a constant C, such that the L_2 norm of the error, e_l, in the lth equation satisfies

$$\|e_l\| \le C\|h^{q_l}\rho_l\|$$

where ρ_l is the residual in the lth equation and q_l is the stability estimate derivative order. The adaptive solver algorithm uses the following L_2-norm error indicator:

$$\left(\int_\Omega \sum_l s_l^{-2} h^{2q_l} |\rho_l|^2 dA\right)^{\frac{1}{2}}$$

with the default value $q_l = 2$. This formula also introduces the scaling factors s_l for the residual with the default value $s_l = 1$.The local error indicator for a mesh element is

$$\sum_l s_l^{-2} h^{2q_l} \tau_l^{\;2} A$$

where A is the area (volume, length) of the mesh element, and τ_l is the absolute value of the lth equation residual (one number per mesh element).

The Functional Error Estimate

The functional-based estimate relies on adjoint solution error estimation. Instead of approximating the error of the solution, the adaptive solver uses the approximation of the error of a certain error functional (Ref. 12). Under rather general assumptions, it is possible to show that the error e (of a functional) can be estimated as

$$|e| \le \sum_l \|e_l^*\| \|\rho_l\|$$

where e_l* and ρ_l are the error in the dual or adjoint solution to, and the residual for, the lth equation, respectively. The adaptive solver algorithm uses the following error indicator for a mesh element:

$$\sum_l w_l \tau_l A$$

where A is the area (volume, length) of the mesh element, and τ_l is the absolute value of the lth equation residual (one number per mesh element). Here w_l is an estimate of the adjoint solution error for the lth equation. This error is estimated in one of two ways. For both methods the sensitivity solver finds the discrete adjoint solution. If only Lagrange element shape functions are used, the solver uses the *ppr* technique to compute w_l as an element average of $|\mathrm{pprint}(u_l^*) - u_l^*|$. Here u_l* is the current computed adjoint solution for the lth equation. If not only Lagrange-element shape functions are used, then $w_l = hD_l$ where D_l is an element average of $|\nabla u_l^*|$. The global error printed in the solver log is the sum of the error indicator for all mesh elements.

The Domain Decomposition Solver

The Domain Decomposition solver or preconditioner is a memory-efficient iterative algorithm for large problems where other methods are infeasible. The basic idea of the iterative (spatial) domain decomposition is as follows.

Consider an elliptic PDE over a domain D and a partition $\{D_i\}$ such that

$$D = \bigcup_i D_i$$

Instead of solving the PDE on the entire D at once, the algorithm solves a number of subdomain problems for each subdomain D_i. If the subdomain D_i is adjacent to a boundary, its boundary conditions are used. On the interfaces between subdomains certain natural transmission conditions arise. It is known (Ref. 21) that the solution to the set of subdomain problems is equivalent to the original problem formulated over D. The solution can be found by iteratively solving each subdomain problem with all other domains fixed. This leads to various domain decomposition methods.

One class of methods is the *overlapping Schwartz method*. There, the partition $\{D_i\}$ is allowed to grow such that each subdomain has a small overlap with its neighboring domains. The size of the overlap is an important parameter that determines the convergence rate for the method. To further accelerate the convergence rate a coarse problem is used. The coarse problem solved on the entire D should yield an estimate of the solution to the full problem on D.

Two practical properties of this method are:

- Control of maximum memory consumption independent of problem formulation: Only a small part of the problem needs to be discretized and solved for at once.
- Coarse-grained concurrency: Disjoint problems can be solved concurrently on different cluster nodes.

Four domain decomposition methods are implemented based on this method: the additive, multiplicative, hybrid, and symmetric Schwartz methods.

The Fully Coupled Attribute and the Double Dogleg Method

For the Fully Coupled attribute, you can define one of the settings for the double dogleg method. Also see Termination Criterion for the Fully Coupled and Segregated Attribute Nodes.

FullyCoupled and Segregated in the *COMSOL API Reference Manual*

THE DOUBLE DOGLEG METHOD

The double dogleg method is available for stationary problems. It is a Newton trust region method and is as such able to adjust the direction as well as the step length when solving the nonlinear equation $F(u) = 0$, F : $R^n \to R^n$.

In order to apply the double dogleg method, consider the minimization of the quadratic model

$$m_k(s) = \frac{1}{2}\|F_k + F'_k s\|^2 = \frac{1}{2}F_k^T F_k + (F'^T_k F_k)^T s + \frac{1}{2}s^T F'^T_k F'_k s$$

subject to $\|s\| \le \delta_k$. Here the size of the step s is required to be bounded by the trust region radius δ_k . Both the Cauchy point—that is, the minimizer of m in the steepest descent direction—as well as the Newton point are utilized. In each iteration the algorithm dynamically adjusts the size of the trust region depending on the predicted decrease of m compared to the actual one. The double dogleg step is then given by a certain convex combination of the Cauchy step (steepest descent direction) and the Newton step. For difficult problems you can choose to start the computation by a damped Newton step. Enter the damping factor between 0 and 1 in the **Initial damping factor** field. The algorithm terminates if the norm of the scaled residual is less than the given tolerance, $\|SF_k\| \le \mathrm{tol}$. You can choose the type of scaling in the **Residual scaling** list. See the Fully Coupled **Method and Termination** settings.

The Iterative Solvers

The following section provides more detailed information about the Iterative solver types—GMRES, FGMRES, Conjugate Gradients, and BiCGStab.

It also discusses the Convergence Criteria for Iterative Solvers and Selecting a Preconditioner for an Iterative Linear System Solver.

Linear in the *COMSOL API Reference Manual*

ITERATIVE SOLVER TYPES

The following information also applies to the Krylov Preconditioner attribute.

These solvers can roughly be ordered according to their memory usage and computational time per iteration (with least memory and time first): Conjugate gradients, BiCGStab, GMRES, and then FGMRES. The solvers that require less memory and computational time per iteration typically are less robust and not applicable to all problem types.

The GMRES Iterative Solver

This linear system solver uses the restarted GMRES (generalized minimum residual) method (see Ref. 8 and Ref. 9). This is an iterative method for general linear systems of the form $Ax = b$. For fast convergence it is important to use an appropriate *preconditioner*.

FGMRES Iterative Solver

This solver uses the restarted FGMRES (flexible generalized minimum residual) method (Ref. 11). The solver is a variant of the GMRES solver that can handle a wider class of preconditioners in a robust way. You can, for example, use any iterative solver as preconditioner for FGMRES. The downside with the method is that it uses twice as much memory as GMRES for the same number of iterations before restart. FGMRES uses right preconditioning and therefore has the same convergence criterion as right-preconditioned GMRES. If FGMRES is used together with a constant preconditioner such as the Incomplete LU preconditioner, then the FGMRES solver is identical to the right preconditioned GMRES solver.

Conjugate Gradients Iterative Solver

This solver uses the conjugate gradients iterative method (Ref. 8, Ref. 12, and Ref. 13). It is an iterative method for linear systems of the form $Ax = b$ where the matrix A is positive definite and (Hermitian) symmetric. Sometimes the solver also works when the matrix is not positive definite, especially if it is close to positive definite. This solver uses less memory and is often faster than the GMRES solver, but it applies to a restricted set of models.

For fast convergence it is important to use an appropriate *preconditioner*, which should be positive definite and (Hermitian) symmetric.

BiCGStab Iterative Solver

This solver uses the *biconjugate gradient stabilized* iterative method (Ref. 8 and Ref. 14) for solving general linear systems of the form $Ax = b$. The required memory and the computational time for one iteration with BiCGStab is constant; that is, the time and memory requirements do not increase with the number of iterations as they do for GMRES. BiCGStab uses approximately the same amount of memory as GMRES uses for two iterations. Therefore, BiCGStab typically uses less memory than GMRES.

The convergence behavior of BiCGStab is often more irregular than that of GMRES. Intermediate residuals can even be orders of magnitude larger than the initial residual, which can affect the numerical accuracy as well as the rate of convergence. If the algorithm detects poor accuracy in the residual or the risk of stagnation, it restarts the iterations with the current solution as initial guess.

In contrast to GMRES and conjugate gradients, BiCGStab uses two matrix-vector multiplications each iteration. This also requires two preconditioning steps in each iteration. Also, when using the left-preconditioned BiCGStab, an additional preconditioning step is required each iteration. That is, left-preconditioned BiCGStab requires a total of three preconditioning steps in each iteration.

CONVERGENCE CRITERIA FOR ITERATIVE SOLVERS

When you use an iterative solver COMSOL estimates the error of the solution while solving. Once the error estimate is small enough, as determined by the convergence criterion

$$\rho\left|M^{-1}(b-Ax)\right| < \mathrm{tol}\cdot\left|M^{-1}b\right| \tag{19-9}$$

the software terminates the computations and returns a solution. When you use a direct solver COMSOL can optionally make a check to determine if the above convergence criterion is fulfilled after the solution step. If the error criterion is not met, the solution process is stopped and an error message is given.

The definitions of M for the various solvers are:

- For MUMPS, PARDISO, and SPOOLES, $M = LU$, where L and U are the LU factors computed by the solver.
- When using left-preconditioning with the iterative solvers GMRES, conjugate gradients, and BiCGStab, M is the preconditioner matrix.
- For the remaining iterative solvers, M is the identity matrix.

The convergence criterion in Equation 19-9 states that the iterations terminate when the relative (preconditioned) residual times the factor ρ is less than a tolerance tol. For solvers where M is equal to the identity matrix, the iterations can sometimes terminate too early with an incorrect solution if the system matrix A is ill-conditioned. For solvers where M is not equal to the identity matrix, the iterations can sometimes terminate too early if M is a poor preconditioner. If the iterations terminate too early due to an ill-conditioned system matrix or a poor preconditioner, increase the factor ρ to a number of the order of the condition number for the matrix $M^{-1}A$. If ρ is greater than the condition number for the matrix $M^{-1}A$, the convergence criterion implies that the relative error is less than tol: $|x - A^{-1}b| < \mathrm{tol}\cdot|A^{-1}b|$.

SELECTING A PRECONDITIONER FOR AN ITERATIVE LINEAR SYSTEM SOLVER

When using an Iterative linear system solver you must select a preconditioner. The choice of preconditioner affects the number of iterations and the solver's eventual convergence. Preconditioning can consume more time and memory than the actual iterative solver itself. To choose a preconditioner, right-click the **Iterative** node and choose one of the following preconditioners from the context menu:

TABLE 19-7: SELECTING A PRECONDITIONER

PRECONDITIONER	USAGE
Incomplete LU	For nonsymmetric systems (the default preconditioner).
Multigrid—Geometric multigrid	For elliptic or parabolic systems.
Multigrid—Algebraic multigrid	For scalar problems or loosely coupled multiphysics problems of the elliptic or parabolic type.
AMS—auxiliary space Maxwell solver	For curl-curl problems stemming from stationary or time-dependent Maxwell's equations.
Jacobi (diagonal scaling)	For large positive definite models.

TABLE 19-7: SELECTING A PRECONDITIONER

PRECONDITIONER	USAGE
SOR	For elliptic problems without zeros on the diagonal. Typically better than Jacobi and still rather inexpensive.
SOR Gauge	For ungauged vector element formulations of Magnetostatics.
SOR Line	For the same problem class as for SOR but adopted to stretched/anisotropic meshes (for example, boundary layer meshes). More expensive than SOR.
SOR Vector	For large electromagnetics models using vector elements.
Krylov Preconditioner	For Helmholtz problems where the mesh does not fulfill the Nyquist criteria. It can be used on the coarse multigrid level or as a smoother.
Vanka	For large indefinite problems with zeros on the diagonal of the system matrix.
SCGS	For fluid-flow problems with linear elements.
Domain Decomposition	For large problems in a distributed-memory system or as an alternative to a direct solver.

Each preconditioner has its own settings; to adjust them, select the preconditioner node to open its settings window. If you want to solve a model without a preconditioner, disable all preconditioner nodes.

The Incomplete LU preconditioner, which is the default preconditioner, works in a more general context than the others, but it might be impractical because of its time and memory requirements; when they work, the multigrid preconditioners are always preferable. The SOR and Jacobi diagonal-scaling preconditioners use less time and memory but only ensure convergence of the iterative solver for positive definite problems. Problems with zeros on the diagonal are efficiently preconditioned with the Vanka preconditioner. To precondition electromagnetic problems that use vector elements for a PDE containing the curl-curl operator, use the SOR Vector preconditioner.

For details about the individual preconditioners, follow the links in the table above.

Preconditioner Selection Guidelines

The physics selects a default preconditioner that is usually appropriate for the problem type, at least for single-physics models. If the default does not perform well, select another one using the following guidelines:

- If the system is elliptic or parabolic (see Elliptic and Parabolic Models) use the geometric multigrid preconditioner.
- If you solve a fluid-flow problem with linear elements only, try the SCGS preconditioner. This is the default setting for most fluid-flow physics.
- If you solve an indefinite problem with zeros on the diagonal of the system matrix, such as the Navier-Stokes equations, try the Vanka preconditioner or the geometric multigrid preconditioner with Vanka or Incomplete LU as the smoother. With appropriate stabilization, it is possible in many cases to use SOR or SOR Line as a GMG pre- and postsmoother instead of Vanka, which increases performance. This is the default setting in some fluid-flow physics.
- If the system is positive definite but so large that the other preconditioners run out of memory, try the SOR Vector as smoother.
- If you solve an electromagnetics problem using vector elements for a PDE containing the curl-curl operator, try the geometric multigrid preconditioner with the SOR vector presmoother and the SOR vector postsmoother, or try the SOR vector preconditioner. Alternatively, if the problem is real-valued stationary or time-dependent you can try the geometric multigrid (GMG) preconditioner with the SOR presmoother and the SOR postsmoother,

and AMS as the coarse solver. AMS is designed for the lowest-order vector elements. For higher-order discretizations use GMG with the option **Lower element order first** and sufficiently number of levels such that AMS could be used efficiently as a coarse solver.

- Try the Incomplete LU preconditioner, which works for all linear systems. However, it requires the tuning of the drop tolerance (or fill ratio); it can run out of memory; and in many cases it is not the most efficient preconditioner.
- If the system is elliptic or parabolic and the system is a real-valued PDE for a single solution component (that is, a scalar problem) you can alternatively try the algebraic multigrid preconditioner.
- As an alternative to the multigrid solver and the use of a direct solver, the Domain Decomposition solver can be a memory efficient alternative and is a scalable solver well suited for use in a distributed memory system.

The Incomplete LU preconditioner and sometimes the multigrid preconditioners require some tuning to get fast convergence without running out of memory (see the sections about these preconditioners).

The Multigrid Solvers

The different Multigrid solvers types—geometric multigrid (GMS) and algebraic multigrid (AMS)—are discussed in this section as well as the multigrid algorithm.

THE GEOMETRIC MULTIGRID SOLVER/PRECONDITIONER

The geometric multigrid solver uses a hierarchy of ***multigrid levels*** where each level corresponds to a mesh and a choice of shape functions. Thus, in addition to coarsening the mesh it is possible to construct a new “coarser” level by lowering the order of the shape functions. The number of degrees of freedom decreases when you go to a coarser multigrid level. The meshes for the different levels can be constructed manually or automatically. The automatic options use a coarsening algorithm to the fine mesh, which leads to meshes that are not aligned to each other. There is also an option to generate the finer meshes from the coarsest mesh by successive mesh refinements, which leads to aligned (nested) meshes. The manual option can be useful when you have a quadrilateral, hexahedral, or prism mesh, or when you for some other reason want to control details in the meshes.

The geometric multigrid solver (GMS) or preconditioner is a fast and memory-efficient iterative method for elliptic and parabolic models. It performs one or several cycles of the geometric multigrid method. The classical multigrid algorithm uses one or several auxiliary meshes that are coarser than the original (fine) mesh. The idea is to perform just a fraction of the computations on the fine mesh. Instead, it performs computations on the coarser meshes when possible, which leads to fewer operations. The size of the extra memory used for the coarser meshes and associated matrices is comparable to the size of the original data. This leads to an iterative algorithm that is both fast and memory efficient. See Ref. 17 for more information.

THE ALGEBRAIC MULTIGRID SOLVER/PRECONDITIONER

The algebraic multigrid solver (AMG) or preconditioner performs one or several cycles of the algebraic multigrid method. This is similar to the geometric multigrid algorithm, the difference being that it constructs the multigrid levels directly from the finest-level system matrix A_0. That is, it constructs the prolongations P_i from A_0 without using auxiliary meshes. It constructs the coarse level matrices A_i from A_0 with the Galerkin projection method. The advantage is that you need not bother about the coarse multigrid levels. The disadvantages are twofold:

- Algebraic multigrid does not work well for vector-valued PDEs in COMSOL’s implementation. That is, it handles only scalar PDEs.
- COMSOL’s implementation does not support complex-valued system matrices.

THE MULTIGRID ALGORITHM

To describe the multigrid algorithm, assume that you have $N+1$ multigrid levels numbered from 0 to N, where 0 is the finest level (the level for which you seek the solution). To solve the linear system $A_0x = b$ (corresponding to level 0), the algorithm must reform the system matrices $A_1, \ldots, A_N$ for the coarse multigrid levels. It must also compute the *prolongation matrices* P_i that map a solution x vector on level i to the corresponding solution vector $P_i x$ on the next finer level $i-1$.

The prolongation matrices are constructed using plain interpolation from one multigrid level to the other. The system matrices for the coarse levels can be constructed in two ways:

- By assembling A_i on the mesh of level i (the default method).
- By projection from the finer level: $A_i = P_i^T A_{i-1} P_i$. This is also called the Galerkin method. It typically leads to more nonzero elements in the system matrix A_i, but the convergence can be faster than in the default method.

The following algorithm describes one multigrid cycle:

1. The input to the algorithm is some initial guess x_0 to the solution of the system $A_0 x = b$.
2. Starting with x_0, apply a few iterations of a *presmoother* to the linear system $A_0 x = b$, yielding a more accurate iterate $x_{0\mathrm{s}}$. Typically the presmoother is some simple iterative algorithm such as SOR, but you can also choose any iterative solver.
3. Compute the residual $r_0 = b - A_0 x_{0\mathrm{s}}$. The presmoother "smooths" the residual so the oscillations in r have such a long wavelength that they are well resolved on the next coarser level (1). Therefore, project the residual onto level 1 by applying the transpose of the prolongation: $r_1 = P_1^T r_0$.
4. If $N = 1$ use the *coarse solver* to solve the system $A_1 x_1 = r_1$. The coarse solver is typically a direct solver such as MUMPS. The number of degrees of freedom on level 1 is less than for level 0, which means that solving $A_1 x_1 = r_1$ is less expensive. If instead $N > 1$, solve the system $A_1 x_1 = r_1$ (approximately) by recursively applying one cycle of the multigrid algorithm for levels $1, 2, \ldots, N$. In both cases the obtained solution x_1 is called the *coarse grid correction*.
5. Map the coarse grid correction to level 0 using the prolongation matrix: $x_{0\mathrm{c}} = x_{0\mathrm{s}} + P_1 x_1$.
6. Starting with $x_{0\mathrm{c}}$, apply a few iterations of a *postsmoother* to the linear system $A_0 x = b$, yielding a more accurate iterate $x_{0\mathrm{mg}}$. The default postsmoother is SORU (the version of SOR using the upper triangle of the matrix). The iterate $x_{0\mathrm{mg}}$ is the output of the multigrid cycle.

The cycle just described is called the V-cycle. The recursive call in step 4 (when $N > 1$) is a also a V-cycle. For the W-cycle and the F-cycle, steps 1–6 above are the same but with the twist that the recursive call in step 4 is substituted with two multigrid calls for the coarser levels. For the W-cycle these two calls are recursive calls (W-cycle calls). For the F-cycle the first call is a W-cycle and the second a V-cycle.

For only two multigrid levels ($N = 1$) these cycles are the same because the algorithm uses the coarse solver in step 4. Also the amount of work on the finest level is the same for the different cycles. Normally the V-cycle is sufficient, but the W-cycle and the F-cycle can be useful for more difficult problems.

When using multigrid as a preconditioner, the action of this preconditioner is obtained by applying a fixed number of multigrid cycles. When using multigrid as a solver, the multigrid cycle repeats until it reaches convergence.

When using multigrid as a preconditioner for the conjugate gradients method for a symmetric matrix A, the preconditioning matrix M should also be symmetric. This requirement is fulfilled if the matrices M associated with the presmoother and the postsmoother are transposes of each other. For instance, this is the case if the presmoother is SOR and the postsmoother is SORU, and if the same number of smoothing steps is used. This combination with two smoothing steps is the default.

Notes on the Efficiency of Smoothers

COMSOL performs smoothing on all but the coarsest multigrid level. A smoother should be computationally cheap and effective at reducing the part of the error that has a high spatial frequency on the mesh to which it is applied. Therefore, applying a smoother on several meshes with a hierarchy of mesh sizes results in a more efficient solver than if the smoother were applied only on the finest mesh.

The efficiency of the multigrid method with simple iterations as a smoother (such as the Jacobi and SOR iteration) hinges on the ellipticity of the underlying mathematical problem. For Helmholtz problems originating from an equation

$$-\nabla\cdot\left(\frac{1}{a}\nabla u\right)-\omega^2 u = f$$

or

$$\nabla\times\left(\frac{1}{a}\nabla\times\mathbf{E}\right)-\omega^2\mathbf{E} = \mathbf{F}$$

the obtained linear problem is indefinite for large frequencies ω. For these problems, a simple iteration amplifies smooth eigenmodes if the mesh is too coarse and makes these methods unsuitable as smoothers. To determine when to use a simple iteration, apply the *Nyquist criterion*

$$h_{\max} < \frac{\lambda}{2} = \frac{\pi}{\omega\sqrt{a}}$$

which says that the mesh must have at least two mesh elements per wavelength. Thus, when using the geometric multigrid solver for these types of problems, ensure that this criterion is fulfilled on the coarsest mesh if simple iterations are used as a smoother. In situations where the criterion is not fulfilled on coarse meshes GMRES can be a suitable smoother (Ref. 20). However, this setting makes smoothers on all levels more expensive and might not always pay off compared to choosing a coarse grid that satisfies the Nyquist criterion. Note also that a smoother based on a Krylov preconditioner like GMRES requires the (outer) iterative solver to be set to FGMRES.

The Parametric Solver Algorithm

The Parametric solver performs a loop around the usual stationary solver in which it estimates the initial guess using the solution for the previous parameter value. If the nonlinear solver does not converge and you are solving for a single parameter, it tries a smaller parameter step; COMSOL determines the size of this step from the convergence speed of the Newton iteration using step-size selection criteria based on work in Ref. 10.

The SCGS Solver

The SCGS iterative solver (smoother) works in a similar way to the blocked update of the Vanka solver, but it builds blocks based on the DOFs in each mesh element instead of blocks based on DOF connectivity to a Vanka variable. The advantage is that the blocks are smaller, allowing for storing their factorization once during the initialization phase (like SOR Line does) instead of factorizing on every update (like Vanka does by default).

Compared to other multigrid smoothers, SCGS provides better performance and is more robust, but it also requires somewhat more memory. SCGS only works for linear elements, and it is the default smoother for fluid-flow models with P1+P1 elements (linear elements for the velocity field and the pressure).

The solver includes three main methods:

- *Mesh elements*: Each mesh element corresponds to one SCGS block.
- *Mesh element lines*: Anisotropic mesh elements are grouped together in SCGS blocks along the direction of anisotropy, which gives better results for boundary layer meshes. Non-anisotropic mesh elements correspond to one SCGS block.
- *Mesh element lines and vertices*: Anisotropic mesh elements are grouped together in SCGS blocks, like above. The DOFs corresponding to non-anisotropic mesh elements are solved using vertex-based SCGS blocks, which consume less memory than element blocks. A separate relaxation factor can be set for the vertex pass.

The solver additionally has an option to use a Vanka hybrid step where Vanka blocks are first built and then SCGS blocks are built excluding the Vanka variable DOFs. This step makes it possible to run iterative solvers when using, for example, the Laminar Inflow boundary condition, independent of whether the mesh is anisotropic or not.

The Segregated Solver

Segregated in the *COMSOL API Reference Manual*

TERMINATION CRITERION FOR A SEGREGATED SOLVER

For the **Solution** termination criterion: When termination of the Segregated solver is based on the estimated error, it terminates if, for all the groups j, the error estimate is smaller than the corresponding tolerance,

$$\mathrm{err}_{j,k} < \mathrm{tol}_j$$

where the error estimate in segregated iteration k is

$$\mathrm{err}_{j,k} = \max(\mathrm{e}^{N}_{j,k}, \mathrm{e}^{S}_{j,k})$$

The number tol_j is the relative tolerance for the corresponding group. For each degree of freedom (DOF) the field variable solved for is $1 \le p \le M$ and $1 \le i \le N_{j,p}$ is the index of that DOF. The largest damped Newton error is then estimated by:

$$\mathrm{e}^{N}_{j,k} = \max_l (1-\alpha_{l,j}) \left[\frac{1}{M} \sum_{p=1}^{M} \frac{1}{N_{j,p}} \sum_{i=1}^{N_{j,p}} \left(\frac{\left|\Delta U_i^{l,j,k,p}\right|}{W_i^{j,p}} \right)^2 \right]^{1/2}$$

Here l is taken for all iterations in all substeps solving for the group j, $\alpha_{l,j}$ is the damping factor, $\Delta U^{l,j,k,p}$ is the Newton increment vector, and $N_{j,p}$ is the number of DOFs in the field p. The weight factor $W_i^{j,p}$ is described below. Moreover,

$$\mathrm{e}^{S}_{j,k} = \left[\frac{1}{M} \sum_{p=1}^{M} \frac{1}{N_{j,p}} \sum_{i=1}^{N_{j,p}} \left(\frac{\left|(U^{j,k,p} - U^{j,k-1,p})_i\right|}{W_i^{j,p}} \right)^2 \right]^{1/2}$$

is the relative increment over one complete iteration k. In this expression, $U^{j,k,p}$ is the segregated solution vector for the group j, and $W_i^{j,p} = \max(\left|U_i^{j,p}\right|, S_i)$, where S_i is a scale factor that the solver determines from the settings in the **Scaling** section of the settings window for the Dependent Variables node, where the following choices are available in the **Method** list:

- For **Automatic**, S_i is the factor 0.1 times the average of $|U_m|$ for all DOFs m having the same name as DOF i.
- For **Manual**, S_i is the value for DOF i given in the **Manual scaling** field.
- For **Initial value based**, S_i is the factor 0.1 times the average of $|U_{0m}|$ for all DOFs m having the same name as DOF i, where U_0 is the solution vector corresponding to the initial value.
- For **None**, $W_i = 1$.

S_i is independent of the field variable p.

For the **Residual** termination criterion, the segregated solver terminates when the following convergence criterion is satisfied: for all the groups j, the error estimate is smaller than the corresponding tolerance, $\mathrm{err}_{j,k} < \mathrm{tol}_j$, where

$$\mathrm{err}_{j,k} = \left[\frac{1}{M}\sum_{p=1}^{M}\frac{1}{N_{j,p}\tilde{W}_{j,p}^{2}}\sum_{i=1}^{N_{j,p}}\left|(F^{j,k,p})_i\right|^2\right]^{1/2}$$

where F is the current residual, and $\tilde{W}$ are the weights determined by the first and, if applicable, the second residual. The iterations can also terminate if the relative solution based error is in the range of a hundred machine epsilon.

Pseudo Time Stepping

Pseudo time stepping is available in a stationary segregated approach as well; see Pseudo Time Stepping for a description of the CFL regulation. For the segregated solver the error estimate e_n in Equation 19-1 is the arithmetic average of the errors in the different segregated groups.

The Sensitivity Analysis Algorithm

When you enable Sensitivity analysis, the stationary solvers compute—in addition to the basic forward solution—the sensitivity of a functional

$$Q = Q(u_p, p) \qquad (19\text{-}10)$$

with respect to the sensitivity variables p. The forward solution u_p is a solution to the parameterized discrete forward problem

$$L(u_p, p) = N_F\Lambda_p \qquad M(u_p, p) = 0 \qquad (19\text{-}11)$$

where Λ_p are the constraint Lagrange multipliers, or (generalized) reaction forces, corresponding to the constraints M. It is assumed that Q does not explicitly depend on Λ_p.

To compute the sensitivity of Q with respect to p, first apply the chain rule:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \frac{\partial Q}{\partial u}\frac{\partial u}{\partial p} \qquad (19\text{-}12)$$

In this expression, the sensitivity of the solution with respect to the sensitivity variables, $\partial u/\partial p$, is still an unknown quantity. Therefore, differentiate the forward problem in Equation 19-11 formally with respect to p:

$$K\frac{\partial u_p}{\partial p} + N_F\frac{\partial \Lambda_p}{\partial p} = \frac{\partial L}{\partial p} + \frac{\partial N_F}{\partial p}\Lambda_p \qquad N\frac{\partial u_p}{\partial p} = \frac{\partial M}{\partial p}$$

Here, $K = -\partial L/\partial u$ and $N = -\partial M/\partial u$ as usual. Assuming that the constraint force Jacobian N_F is independent of p (that is, $\partial N_F/\partial p = 0$), you can write the above relations in matrix form

$$J\begin{pmatrix}\dfrac{\partial u_p}{\partial p}\\[2mm] \dfrac{\partial \Lambda_p}{\partial p}\end{pmatrix} = \begin{pmatrix}\dfrac{\partial L}{\partial p}\\[2mm] \dfrac{\partial M}{\partial p}\end{pmatrix} \qquad J = \begin{bmatrix}K & N_F\\ N & 0\end{bmatrix} \qquad (19\text{-}13)$$

Solve for the sensitivities $\partial u_p/\partial p$ and $\partial \Lambda_p/\partial p$, and plug them back into Equation 19-12:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \begin{pmatrix}\frac{\partial Q}{\partial u}\\ 0\end{pmatrix}^T J^{-1}\begin{pmatrix}\frac{\partial L}{\partial p}\\ \frac{\partial M}{\partial p}\end{pmatrix} \qquad (19\text{-}14)$$

This formula gives dQ/dp explicitly in terms of known quantities, but in practice it is too expensive to invert the matrix J.

If the number of individual sensitivity variables, p_j, is small, Equation 19-13 can be solved for each right-hand side $[\partial L/\partial p_j\ \partial \mathrm{M}/\partial p_j]^T$, and the solution is then inserted into Equation 19-12. This is the *forward method*, which in addition to the sensitivity dQ/dp returns the sensitivity of the solution, $\partial u_p/\partial p$. The matrix J is in fact the same matrix as in the last linearization of the forward problem. The forward method therefore requires one additional back-substitution for each sensitivity variable.

If there are many sensitivity variables and the sensitivity of the solution itself, $\partial u_p/\partial p$, is not required, the *adjoint method* is more efficient. It is based on using auxiliary variables $u*$ and $L*$, known as the *adjoint solution*, to rewrite Equation 19-14:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \begin{pmatrix}u*\\ \Lambda*\end{pmatrix}^T\begin{pmatrix}\frac{\partial L}{\partial p}\\ \frac{\partial M}{\partial p}\end{pmatrix}$$

$$J^T\begin{pmatrix}u*\\ \Lambda*\end{pmatrix} = \begin{pmatrix}\frac{\partial Q}{\partial u}\\ 0\end{pmatrix}$$

In this form only one linear system of equations must be solved regardless of the number of sensitivity variables, followed by a simple scalar product for each variable. This is much faster than the forward method if the number of variables is large. The system matrix, which is solved for, is the transpose of the last linearization of the forward problem. This makes no significant difference for the iterative linear solvers. For the direct solvers, if J is symmetric or Hermitian, this makes no difference compared to the forward method, and the direct solvers can reuse the factorization. In the nonsymmetric case, MUMPS and PARDISO can reuse the factorization of J while SPOOLES needs to do a new factorization of J^{T}.

Sensitivity in the *COMSOL API Reference Manual*

About the SOR, SOR Gauge, SOR Line, and SOR Vector Iterative Solver Algorithms

The background information for the SOR, SOR Gauge, SOR Line, and SOR Vector attribute nodes are described in this section.

THE SOR METHOD

The SOR (successive over-relaxation) method provides a simple and memory-efficient solver/preconditioner/smoother based on classical iteration methods for solving a linear system $Ax = b$. Given a relaxation factor ω (usually between 0 and 2), a sweep of the SOR method transforms an initial guess x_0 to an improved approximation $x_1 = x_0 + M^{-1}(b - Ax_0)$, where the preconditioning matrix $M = L + D/\omega$, and D is the diagonal part of A, and L is the strictly lower triangular part of A. When $\omega = 1$ (the default), the Gauss-Seidel method is obtained.

In the SORU method, $M = U + D/\omega$, where U is the strictly upper triangular part of A. The SOR and SORU methods use a more accurate approximation of the matrix, which leads to fewer iterations but slightly more work per iteration than in the Jacobi method.

The SSOR (symmetric successive over-relaxation) method is one SOR sweep followed by a SORU sweep. The output x_1 for an input x_0 also comes from the above formula but with

$$M = \frac{\omega}{2-\omega}\left(L + \frac{D}{\omega}\right)D^{-1}\left(U + \frac{D}{\omega}\right)$$

When the system matrix A is symmetric, the SSOR method has the advantage that the preconditioning matrix M is symmetric. Symmetry of the preconditioner matrix is necessary when using the conjugate gradients iterative method. In such cases, the SSOR preconditioner is preferable to the SOR preconditioner.

THE SSOR GAUGE, SOR GAUGE, AND SORU GAUGE ALGORITHMS

The SOR Gauge algorithms are described.

Magnetostatic problems are often formulated in terms of a magnetic vector potential. The solution of problems formulated with such a potential is in general not unique. Infinitely many vector potentials result in the same magnetic field, which typically is the quantity of interest. A finite element discretization of such a problem results in a singular linear system of equations, $Ax = b$. Despite being singular, these systems can be solved using iterative solvers if the right-hand side of the discretized problem is the range of the matrix A. For discretized magnetostatic problems, the range of A consists of all divergence-free vectors. Even if the right side of the mathematical problem is divergence free, the right side of the finite element discretization might not be numerically divergence free. To ensure that b is in the range of A, SOR gauge performs a divergence cleaning of the right side by using the matrices T and T^T similar to the algorithm for the SOR Vector iterative method. To this end, the system $T^TT\psi = -T^Tb$ is first solved. Adding $T\psi$ to b then makes the numerical divergence of the right side small.

THE SOR LINE ALGORITHM

In regions where the mesh is sufficiently anisotropic, the algorithm forms *lines* of nodes (SOR Line) that connect nodes that are relatively close to each other (Ref. 27). Thus, in a boundary layer, a line is a curve along the thin direction of the mesh elements. A smoothing iteration does two things:

- Line update: Performs block SOR smoothing where each block consists of degrees of freedom located on a line. Due to the banded structure of each block matrix, this smoothing runs relatively fast.
- SSOR update: Performs a number of SSOR smoothing iterations on the whole mesh.

Like the SOR and Jacobi smoothers/preconditioners, the algorithm gives an error message if it finds zeros on the diagonal of the system matrix.

THE SOR VECTOR ALGORITHM

The SOR Vector algorithm is an implementation of the concepts in Ref. 24 and Ref. 18. The algorithm applies SOR iterations on the main linear equation $Ax = b$ but also makes SOR iterations on a projected linear equation $T^TATy = T^Tb$. Here the projection matrix, T, is the discrete gradient operator, which takes values of a scalar field in the mesh vertices and computes the vector-element representation of its gradient. Loosely speaking, the argument for using this projection is the following: For example, let the linear equation $Ax = b$ represent the discretization of a PDE problem originating from the vector Helmholtz equation

$$\nabla\times(a\nabla\times\mathbf{E}) + c\mathbf{E} = \mathbf{F}$$

for the unknown vector field $\mathbf{E}$, where a and c are scalars, and $\mathbf{F}$ is some right-hand side vector. Standard preconditioners/smoothers cannot smooth the error in the null space of the operator $\nabla \times (a\nabla \times .)$. This null space is the range of the gradient operator. This algorithm adds a correction $\mathbf{E} \to \mathbf{E} + \nabla\phi$ to the standard SOR smoothed solution (or residual), where it computes ϕ from SOR iterations on a projected auxiliary problem. The projected problem is obtained by taking the divergence (or discretely $-T^T$) of the Helmholtz equation and plugging in the correction. You then obtain

$$-\nabla \cdot (c\nabla\phi) = -\nabla \cdot \mathbf{F}$$

(for clarity, boundary constraints are disregarded), which, if c is definite (strictly positive or strictly negative), is a standard elliptic type of equation for the scalar field ϕ .

When using this algorithm as a smoother for the multigrid solver/preconditioner, it is important—for the correct discrete properties of the projected problem—to generate nested meshes. Also, it performs an element assembly on all mesh levels (controlled by the multigrid **Assemble on all levels** check box). You can generate nested meshes through manual mesh refinements or do so automatically by selecting **Refine mesh** from the **Hierarchy generation method** list in the **Multigrid** node.

The projection matrix T is computed in such a way that nonvector shape functions are disregarded, and you can therefore use it in a multiphysics setting. It can also handle contributions from different geometries. Non-vector shape function variables are not affected by the correction from the projected system, and the effects on them are the same as when you apply the standard SOR algorithm.

The Vanka Algorithm

The algorithm is a local smoother/preconditioner of Vanka type. It is based on the ideas in Ref. 19, Ref. 26, and Ref. 27. It is possible to describe it as a block SOR method, where the local coupling of the degrees of freedom (DOFs) determines the blocks. The important idea in this algorithm is to use Lagrange multiplier variables to form the blocks. For illustration purposes, consider the Navier-Stokes equations. For these equations the pressure variable plays the role of Lagrange multiplier. The linearized equations on discrete form has the following structure:

$$A\begin{bmatrix}U\\P\end{bmatrix} = \begin{bmatrix}S & D^T\\D & 0\end{bmatrix}\begin{bmatrix}U\\P\end{bmatrix} = \begin{bmatrix}F\\G\end{bmatrix}$$

where U and P are the velocity and pressure degrees of freedom, respectively. The algorithm loops over the Lagrange multiplier variable DOFs (here the pressure DOFs P_j) and finds the direct connectivity to this DOF. To do so, the algorithm locates the nonzero entries in the matrix column corresponding to P_j. The row indices of the nonzero entries define the DOFs U_k, and the algorithm forms a local block matrix based on this connectivity:

$$A_j = \begin{bmatrix}S_j & D_j^T\\D_j & 0\end{bmatrix}$$

One *Vanka update* loops over all P_j and updates

$$\begin{bmatrix}U_j\\P_j\end{bmatrix} \leftarrow \begin{bmatrix}U_j\\P_j\end{bmatrix} + \omega A_j^{-1}\left(\begin{bmatrix}F\\G\end{bmatrix} - A\begin{bmatrix}U\\P\end{bmatrix}\right)_j$$

where the $(.)_j$ denotes the restriction of a vector to the rows corresponding to the block j. ω is a relaxation parameter. The algorithm does not form the inverses of the block matrices explicitly. Instead, it computes the Vanka update with a LAPACK direct solver or a GMRES iterative method subroutine call. The GMRES method is the restarted GMRES without preconditioning. The algorithm relies on the fact that it is possible to invert the submatrices A_j. If it is not possible, the algorithm gives an error message. A zero on the diagonal of A or A_j is not necessarily a problem for this updating strategy. In general, the Vanka update does not necessarily update all DOFs. This is the case for problems with weak constraints, where only a small subset of the problem's DOFs are directly coupled to the Lagrange multipliers for the constraints. Another example is the Navier-Stokes equations coupled to other equations, but where the coupling is not directly through the pressure variable (in the k-ε turbulence model, for example). The Vanka algorithm automatically detects DOFs that are not updated by the above Vanka updating procedure and performs, for each Vanka update, a number of SSOR sweeps for these DOFs. This part of the algorithm is the *SSOR update*; it only works for a submatrix that has a nonzero diagonal. Just as the SOR and Jacobi preconditioner algorithms, this algorithm gives an error message if it finds zeros on the diagonal for the DOFs in the SSOR update.

Adaptive Mesh Refinement

The **Adaptive Mesh Refinement** node ( ) is an attribute that handles adaptive mesh refinement together with a time-dependent solver. The adaptive mesh refinement then creates multiple meshes for segments of the time-dependent simulation. You can add it together with the Eigenvalue Solver, Stationary Solver, and Time-Dependent Solver operation nodes. Also see The Adaptive Mesh Refinement Solver.

GENERAL

The software performs adaptive mesh refinement in one geometry only. Use the **Adaption in geometry** list to specify that geometry.

Use the **Maximum number of elements** field to specify the maximum number of elements in the refined mesh. If the number of elements exceeds this number, the solver stops even if has not reached the number specified in the **Maximum number of refinements** field.

General Settings for the Stationary and Eigenvalue Adaptive Solver

Use the **Maximum number of refinements** field to specify the maximum number of mesh refinement iterations. The default value is 2 refinements.

General Settings for the Time-Dependent Adaptive Solver

The following properties appear under **Time interval control**:

- The **Time interval length** can be controlled manually or automatically. Select **Manual** (default) or **Automatic**.
- The value in the **Interval reduction factor** field (default value: 0.5) determines how the solver reduces the time interval length. A value of 0.5 makes the interval length half of the previous interval length when reduced.
- By default, the solver determines the **Interval length** (unit: s) automatically (only available when **Time interval length** is **Manual**) using an interval length that gives a total of 10 intervals. The length of the time interval is the simulation time before a refinement of the mesh takes place. Click the check box to specify a user-defined time interval length in the field (default value: 0.1 s).
- The value in the **Interval growth factor** (only available when **Time interval length** is **Automatic**) edit field (default value: 2.0) determines how the solver increases the time interval length. A value of 2.0 makes the interval length twice as large as the previous interval length when increased.
- By default, the solver determines the **Initial interval length** (unit: s) automatically (only available when **Time interval length** is **Automatic**). The length of the initial time interval is the simulation time before the first refinement of the mesh takes place. Select the check box to specify a user-defined time interval length in the field (default value: 0.1 s).
- By default, the solver also determines the **Minimal interval length** (unit: s) automatically. The minimal length of the time interval is the shortest possible simulation time without performing a mesh refinement. Click to select the check box to specify a user-defined minimal interval length in the field (default value: 0.01 s).
- If **Time interval length** is **Automatic** the algorithm strives to assume the given value of Fraction of maximum refinement (default value: 0.2) by controlling the size of the time interval. A value of zero means no refinement of the base mesh and a value of one means refinement everywhere using Maximum element refinements. The

shortening and lengthening of the interval is determined by the interval reduction and growth factors described below.

For the properties under **Mesh element control**; see Mesh Refinement below. Also, select or clear the **Convert to simplex mesh** check box (the default is to use this conversion). Mesh refinement is only possible for simplex meshes. If the original mesh is not a simplex mesh it can be converted to a simplex mesh by this selection.

ERROR ESTIMATION

Use the **Error estimate** list to control how the error estimate is computed.

Error Estimation for the Stationary and Eigenvalue Solvers

Select **L2 norm** to use the L_2 norm.

Select **Functional** to specify a globally available scalar-valued expression. This option adapts the mesh toward improved accuracy in the expression for the functional. This is only available when using with the Stationary Solver.

Further options regarding error estimation (available as indicated for each option) are:

- **Scaling factor** (only available when **Error estimate** is **L2 norm**). Use this field to enter a space-separated list of scaling factors s_l, one for each field variable (default: 1). The error estimate for each field variable is divided by this factor.
- **Stability estimate derivative order** (only available when **Error estimate** is **L2 norm**). The L_2 norm error estimate is based on a stability estimate for the PDE. Use this field to specify its order.
- **Functional** (only available when **Error estimate** is **Functional**). Use this edit field to specify a globally available scalar-valued expression to be used for the error estimate.
- **Adjoint solution error estimate** (only available when **Error estimate** is **Functional**). Use this list to select error estimate method in the adjoint solution: a recovery technique and a gradient-based method. Select **On** to enforce using the recovery technique, and select **Off** to use the gradient-based method. Select **Automatic** to let the solver check if the geometry only contains Lagrange basis functions. If so, the adjoint solution uses the recovery technique. Otherwise, it uses the gradient-based method.
- **Weights for eigenmodes** (only available with the Eigenvalue Solver). For eigenvalue problems, the error estimate is a weighted sum of the error estimates for the various eigenmodes. Use this field to enter a space-separated list of positive (relative) weights. The default value of 1 means that all the weight is put on the first eigenmode.

Error Estimation for the Time-Dependent Solver

For time-dependent adaptive mesh refinement you need to specify a user-defined **Error indicator**. Use the edit field to give the error indicator function used for the adaptive mesh refinement. For background information to help with this section, see The Adaptive Mesh Refinement Solver (Error Estimates).

MESH REFINEMENT

Use the **Refinement method** list to control how to refine mesh elements. Select:

- **Longest** to make the solver refine only the longest edge of an element. (Not available for 1D geometries.)
- **Mesh initialization** to generate a completely new mesh. (Not available for time-dependent adaptive mesh refinement.)
- **Regular** to make the solver refine elements in a regular pattern. (Not available for 3D geometries.)

By default, the software automatically determines the order of decrease in equation residuals on basis of the shape function orders in the geometry. To specify a residual order manually, select the **Residual order** check box and specify a nonnegative integer in the accompanying field. This option is not available for time-dependent adaptive mesh refinement.

Use the **Element selection** list to specify how the solver should select which elements to refine. Select:

- **Rough global minimum** to minimize the L_2 norm of the error by refining a fraction of the elements with the largest error in such a way that the total number of elements increases roughly by the factor specified in the accompanying **Element growth rate** field. The default value is 1.7, which means that number of elements increases by about 70%.
- **Fraction of worst error** to refine elements whose local error indicator is larger that a given fraction of the largest local error indicator. Use the accompanying **Element fraction** field to specify the fraction. The default value is 0.5, which means that the fraction contains the elements with more than 50% or the largest local error.
- **Fraction of elements** to refine a given fraction of the elements. Use the accompanying **Element fraction** field to specify the fraction. The default value is 0.5, which means that the solver refines about 50% of the elements.

For time-dependent adaptive mesh refinement, you can specify the maximum number of refinements of the mesh elements (default: 2) in the **Maximum element refinements** field.

RESTART

This section is available for the Time-Dependent Solver. After each mesh adaption the time integration is restarted and you can control the following time-stepping properties:

By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.

Use the **Consistent initialization** list to control how the solver performs consistent initialization of differential-algebraic systems by selecting **Off** (the default), **On**, or **Backward Euler**.

OUTPUT

This section contains information about the solution and mesh that contain the results from the adaptive mesh refinement.

PLOT WHILE SOLVING

This section is available for the Stationary Solver and Eigenvalue Solver.

To plot the adaptive mesh refinement, select the **Plot while solving** check box. You can then select any existing plot group from the **Plot group** list to use for the plot.

- The Adaptive Mesh Refinement Solver
- Adaptive Mesh Refinement (Utility Node)
- The Log Window

Implementing a Point Source: model library path **COMSOL_Multiphysics/Equation-Based_Models/point_source**.

Advanced

The **Advanced** node ( ) is an attribute node that handles advanced settings for solver nodes such as a Stationary Solver, Eigenvalue Solver, and Time-Dependent Solver. Also see About the Advanced Attribute Settings.

GENERAL

Matrix Symmetry

Use the **Matrix symmetry** list to control how the solver handles matrix symmetry of linear system matrices. Select:

- **Automatic** to perform automatic symmetry detection. Both symmetric and Hermitian matrices can be detected.
- **Nonsymmetric** to override the automatic symmetry detection and force the solver to assume that matrices are nonsymmetric.
- **Symmetric** to override the automatic symmetry detection and force the solver to assume that matrices are symmetric.

> ⚠ Selecting **Symmetric** for a problem that does not result in symmetric matrices leads to an incorrect solution. See Which Problems are Symmetric? for information.

- **Hermitian** to override the automatic symmetry detection and force the solver to assume that matrices are Hermitian.

Matrix Format

The matrix format can be a s*parse matrix* or a *full matrix* or, which you choose as **Sparse** or **Filled**, respectively, from the **Matrix format** list. The default format is **Automatic**; this setting chooses a sparse or filled matrix format based on the solver used.

Row Equilibration

Even if variables are well scaled, equations can have very different scales. The **Row equilibration** check box is selected to balance the equations using row equilibration. Even when this check box is selected, row equilibration is not used in the following situations in order to preserve matrix symmetry:

- Automatic matrix symmetry detection is used and the system matrices are symmetric
- **Symmetric** or **Hermitian** is selected in the **Matrix symmetry** list
- The conjugate gradients or geometric multigrid solver is used
- The eigenvalue solver is used.

Null-space Function

Use the **Null-space function** list to select a method for the computation of matrices needed for constraint handling (see Elimination Constraint Handling). Select:

- **Automatic** to let the software automatically determine the most appropriate method (**Orthonormal** or **Sparse**).
- **Orthonormal** to compute the needed matrices using singular value decomposition.
- **Sparse** to handle constraint matrices with nonlocal couplings using a sparse algorithm.

Assembly Block Size

By default, the solver chooses the number of mesh elements that are processed together during the assembly process (the block size). Select the **Assembly block size** check box for manual specification of a block size.

Solver Log

The solver log contains information about the progress and convergence of the solvers (see The Log Window). From the **Solver log** list, choose one of the following formats:

- **Minimal**, which reports a minimal amount of information (for example, warnings and nonstandard feedback). The normal output (once per “step” and so forth) is turned off.

- **Normal** (the default), which reports information about the main solver (the time-dependent solver, for example).
- **Detailed**, which reports information about the main solver and also information about the solver on the level below (a nonlinear stationary solver, for example).

Other Settings

Select the **Allow complex-valued output from functions with real input** check box to control whether the solver treats such complex-valued output as an error or not.

The **Stop when undefined mathematical operation is detected** check box controls how the solver handles undefined mathematical operations such as division by zero.

REASSEMBLY

Select the **Manual control of reassembly** check box to be able to override the solver mechanism that automatically detects which quantities need to be reassembled. This can be useful to improve efficiency in situations when the automatic mechanism is too sensitive and reassembles quantities that do not need to be reassembled.

Constant Load

If the **Manual control of reassembly** check box is selected, the **Constant load** is **On** by default.

The load (residual vector) is constant if the PDE and the Neumann boundary conditions are linear with time-independent coefficients and right-hand sides. For the discretized model, this means that the residual vector L depends linearly on U:

$$L = L_0 - KU - D\dot{U} - E\ddot{U}$$

and that L_0, K, D, and the mass matrix E are constant.

If you choose to turn **Off** the **Constant load**, it instructs the solver to perform a reassembly process for the computation of the residual vector (when **Off**) or not. However, even if it is off, you might still want to treat some of the matrices as constant. Manual control of reassembly of these quantities can be controlled with the available check boxes, which makes the assembly only occur once for the corresponding matrix.

- Select the **Constant stiffness** check box to treat the stiffness matrix K as constant.
- Select the **Constant damping or mass** check box if you want to treat the coefficients of the first-order time-derivative terms or the second-order time-derivative terms as constant. In the discretized model, this means that the damping (sometimes called mass) matrix D or the mass matrix E is treated as constant.
- Select the **Constant mass** check box to treat the mass matrix E as constant.

Constant Constraint

If the **Manual control of reassembly** check box is selected you can control reassembly of the constraint residual. By default the **Constant constraint** is **On**. The constraint is constant if the Dirichlet boundary conditions (constraints) are linear and time independent. For the discretized model, this means that the constraint residual M depends linearly on U ($M = M_0 - NU$) and that M_0 and N are constant. It is also assumed that the constraint Jacobian N is correct.

If you choose to turn **Off** the **Constant constraint**, it instructs the solver to perform a reassembly process for the computation of the constraint residual vector (when **Off**) or not. However, even if it is off, the constraint Jacobian might still be constant. To control the reassembly of this quantity select the **Constant constraint Jacobian** check box if the Dirichlet boundary conditions are linear with time-independent coefficients (not right-hand side). For the discretized model this means that N is constant.

AMS

The **AMS** node ( ) is an attribute that handles parameters for linear system solvers/preconditioners that use the auxiliary space Maxwell solver (AMS). Right-click an Iterative, Krylov Preconditioner, or Coarse Solver node to add an **AMS** node.

The AMS solver uses the auxiliary space Maxwell solver preconditioner from the Lawrence Livermore National Laboratory linear solver/preconditioner library *hypre*, a software library of high performance preconditioners and solvers (Ref. 6). AMS provides edge finite element discretization of variational curl-curl problem stemming from stationary or time-dependent Maxwell's equations. The version of AMS available in COMSOL Multiphysics is designed for the lowest-order edge elements. For higher-order discretizations, use it together with the geometric multigrid (GMG) solver with the option **Lower element order first** and sufficiently number of levels so that AMS can work efficiently as a coarse solver. For details, see Ref. 7.

The settings window contains the following section:

GENERAL

Enter the **Number of iterations** of the AMS solver. The default is 2.

In the **Variables** field, add the applicable dependent variables that use vector elements (such as magnetic scalar potential) and that you want to include in the AMS solver. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✕), and **Add** (+) buttons to configure the list of variables.

From the **Cycle type** list, select one of the available AMS cycle types 1–14 (the default is cycle type 1, a multiplicative solver that should work well in most cases; see Ref. 7 for details). These cycle types are various combinations of smoothing and applications of algebraic multigrid on decomposed problems.

From the **Magnetostatics** list, select **Automatic** (the default), **On**, or **Off**. The automatic case determines magnetostatics by comparing the maximum row sum of absolute values for the projected matrix T^TAT and A. Here T is the discrete gradient matrix; see documentation for SOR Vector. Magnetostatics is deduced if the projected matrix is negligible compared to A. If magnetostatics is deduced or chosen, AMS skips the subspace corrections associated with the projected matrix T^TAT.

From the **Divergence cleaning** list, select **Automatic** (the default), **On**, or **Off**. The automatic case is the same as for the determination of magnetostatics. In the magnetostatic case, AMS should skip corrections associated with T^TAT and use divergence cleaning of the right-hand side. You can also manually specify the magnetostatics and divergence cleaning settings. This can be useful if divergence cleaning has already been made or if you suspect that the automatic detection fails.

Automatic Remeshing

The **Automatic Remeshing** node ( ) is an attribute that adds automatic remeshing. The remeshing occurs when the mesh quality falls below a specified value. It can be used together with the Moving Mesh Interface to assure a satisfactory mesh quality throughout the simulation. Right-click a Time-Dependent Solver operation node to add it.

AutoRemesh in the *COMSOL API Reference Manual*

GENERAL

The software only performs automatic remeshing in one geometry. Use the **Remesh in geometry** list to specify that geometry if the model contains more than one geometry.

CONDITIONS FOR REMESHING

From the **Condition type** list choose between different types of conditions for when remeshing should occur. Select:

- **Mesh quality** (the default). The solver remeshes when the mesh quality becomes less than a given limit. Edit the **Mesh quality expression** as required, or click the **Replace Expression** button ( ) to choose another expression. Edit the limit in the **Stop when mesh quality is below** field (default value: 0.2).
- **Distortion**. The solver remeshes when the distortion the mesh has undergone becomes larger than a given limit. Edit the **Distortion expression** or click the **Replace Expression** button ( ) and choose another expression. Edit the limit in the **Stop when distortion exceeds** field (default: 2).
- **General**. The solver remeshes when a logical condition becomes true. Edit the condition in the **Stop when condition is true** field or click the **Replace Expression** button ( ) and choose another expression.

The **Remesh at** setting determines which previous solution is used for the remeshing:

- When **Last output from solver before stop** is selected, the remeshing is done on the last solution that would have been stored by the solver if remeshing would not have occurred. This setting discards any solver progress done since the last output.
- When **Last step taken by solver before stop** is selected (the default), the remeshing is done using the solution from the last solver step before the condition for remeshing became fulfilled. Only the very last solver step, at which the condition was triggered, is discarded. Typically this setting is preferred because then the progress of the automatic remeshing does not depend on the solver's list of output times.

REMESH

After each remeshing, the time integration is restarted and you can control the following time-stepping properties.

By default the solver chooses an initial step automatically. Select the **Initial step** check box to enter a different value (SI unit: s).

To control how the solver performs **Consistent initialization** of differential-algebraic systems, select **Off** (the default), **On**, or **Backward Euler** from the list.

OUTPUT

This section contains information about which solution and meshes contain the results from the automatic remeshing node.

Coarse Solver

The **Coarse Solver** node ( ) is an auxiliary attribute subnode used by the Multigrid and Domain Decomposition attribute nodes. This node does not have any settings. Instead, its purpose is to administrate coarse grid solvers for a multigrid solver. To add a solver, right-click the **Coarse Solver** node.

Control Field

The **Control Field** node is an attribute node that handles settings for field variables that are acting as control variables. Control variables have a special status when using the Sensitivity or Optimization solver. Each control field has a separate **Control Field** node. This attribute is used together with the Dependent Variables node.

GENERAL

The **Field components** section displays the variable names for the fields components.

Use the **Solve for this field** check box to control whether to use this variable when solving a sensitivity or optimization problem. For other parts of the solution process, the control fields are held fixed. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for, its values are determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.

A variable can still be solved for despite not being stored in output and vice versa.

Control State

The **Control State** ( ) node is an attribute node that handles settings for state variables that are acting as control variables. Control (state) variables have a special status when using the Sensitivity or Optimization solver. Each control state has a separate **Control State** node. This attribute is used together with the Dependent Variables node.

GENERAL

The **State Components** section displays the variable names for the states components.

Use the **Solve for this state** check box to control whether to use this variable when solving a Sensitivity or Optimization problem. For other parts of the solution process, the control variables are held fixed. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for its value is determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.

A variable can still be solved for despite not being stored in output and vice versa.

Direct

The **Direct** node ( ) is an attribute that handles settings for direct linear system solvers. Use it together with a Stationary Solver, Eigenvalue Solver, and Time-Dependent Solver, for example. The attribute can also be used together with the Coarse Solver attribute when using multigrid linear system solvers.

An alternative to the direct linear system solvers is given by iterative linear system solvers which are handled via the Iterative attribute node. Several attribute nodes for solving linear systems can be attached to an operation node, but only one can be active at any given time.

Also see Choosing the Right Linear System Solver, which describes The MUMPS Solver, The PARDISO Solver, and The SPOOLES Solver.

GENERAL

Select a linear system **Solver**. Select:

- **MUMPS** (multifrontal massively parallel sparse direct solver) (the default).
- **PARDISO** (parallel sparse direct solver). See for more information about this solver.
- **SPOOLES** (sparse object oriented linear equations solver).
- **Dense matrix** to use a dense matrix solver. The dense matrix solver stores the LU factors in a filled matrix format. It is mainly useful for boundary element (BEM) computations.

MUMPS

If **MUMPS** is chosen, MUMPS estimates how much memory the unpivoted system requires. Enter a **Memory allocation factor** to tell MUMPS how much more memory the pivoted system requires. The default is 1.2.

Select a **Preordering algorithm**—**Automatic** (the default automatically selected by the MUMPS solver), **Approximate minimum degree**, **Approximate minimum fill**, **Quasi-dense approximate minimum degree**, **Nested dissection**, or **Distributed nested dissection**.

Select the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not. Click to clear the check box to turn off the weight matching strategy.

The default **Use pivoting** is **On**, which controls whether or not pivoting should be used.

- If the default is kept (**On**), enter a **Pivot threshold** number between 0 and 1. The default is 0.1. This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column.
- If **Off** is chosen, enter a value for the **Pivoting perturbation**, which controls the minimum size of pivots (the pivot threshold). The default is 10^{-8}.

Select the **Out-of-core** check box to store matrix factorizations (LU factors) as blocks on disk rather than in the computer's internal memory. When the check box is selected, you control the maximum amount of internal memory allowed for the blocks by entering a value in the **In-core memory (MB)** field. The default is 512 and it stores the blocks in RAM and not on disk. The MUMPS out-of-core solver stores the LU factors on the hard drive. This minimizes the internal memory usage. The cost is longer solution times because it takes longer time to read and write to disk than using the internal memory.

You can specify the temporary directory where MUMPS stores the LU factors using the `-tmpdir` switch. See Running COMSOL.

PARDISO

Select a **Preordering algorithm**—**Nested dissection multithreaded** (the default to perform the nested dissection faster when COMSOL runs multithreaded), **Minimum degree**, or **Nested dissection**.

Select a **Scheduling method** to use when factorizing the matrix:

- **Auto** (the default): Selects one of the two algorithms based on the type of matrix.
- **One-level**
- **Two-level**: Choose this when you have many cores as it is usually faster.

Select the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not. Click to clear the check box to turn off the weight matching strategy.

By default the **Bunch-Kaufmann** check box is not selected. Click to select and control whether to use Bunch-Kaufmann pivoting.

By default the **Multithreaded forward and backward solve** check box is not selected. Click to select and run the backward and forward solves multithreaded. This mainly improves performance when there are many cores and the problem is solved several times, such as in eigenvalue computations and iterative methods.

The **Pivoting perturbation** field controls the minimum size of pivots (the pivot threshold ε).

To avoid pivoting, PARDISO uses a pivot perturbation strategy that tests the magnitude of the potential pivot against a constant threshold of $\varepsilon = \alpha|PP_{\mathrm{MPS}}D_rAD_cP|_\infty$, where P and P_{MPS} are permutation matrices, D_r and D_c are diagonal scaling matrices, and $|\cdot|_\infty$ is the infinity norm (maximum norm). If the solver encounters a tiny pivot during elimination, it sets it to $\mathrm{sign}(l_{ii})\varepsilon|PP_{\mathrm{MPS}}D_rAD_cP|_\infty$. The perturbation strategy is not as robust as ordinary pivoting. In order to improve the solution PARDISO uses iterative refinements.

Select the **Out-of-core** check box to store matrix factorizations (LU factors) as blocks on disk rather than in the computer's internal memory. When the check box is selected, you control the maximum amount of internal memory allowed for the blocks by entering a value in the **In-core memory (MB)** field. The default is 512 and it stores the blocks in RAM and not on disk. The PARDISO out-of-core solver stores the LU factors on the hard drive. This minimizes the internal memory usage. The cost is longer solution times because it takes longer time to read and write to disk than using the internal memory.

You can specify the temporary directory where PARDISO stores the LU factors using the `-tmpdir` switch. See Running COMSOL.

SPOOLES

Select a **Preordering algorithm**—**Best of ND and MS** (the best of nested dissection and multisection), **Minimum degree**, **Multisection**, or **Nested dissection**.

Enter a **Pivot threshold** number between 0 and 1. The default is 0.1. This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column.

ERROR

The accuracy of the solution of the linear system is controlled from the **Check error estimate** list. The default is **Automatic**, meaning the main solver is responsible for error management. However, to avoid false termination, the main solver continues iterating until the error check passes or until the step size is smaller than about $2.2\cdot10^{-14}$. Or choose **No** for no error checking or **Yes** for error checking, where no exceptions are thrown by the direct solver. If an error occurs in the main solver, warnings originating from the error checking in the direct solver appear. The error check asserts that the relative error times a stability constant ρ is sufficiently small. Use the **Factor in error estimate** field to manually set ρ. The default is 400. To avoid false termination, the main solver continues iterating until the error check passes or until the step size is smaller than about $2.2\cdot10^{-14}$.

For an example using a Stationary Solver, The Blasius Boundary Layer: model library path **COMSOL_Multiphysics/Fluid_Dynamics/blasius_boundary_layer**.

For an example using an Eigenvalue Solver, Isospectral Drums: model library path **COMSOL_Multiphysics/Equation-Based_Models/isospectral_drums**.

Domain Decomposition

The **Domain Decomposition** node () is an attribute that can be used together with the Iterative, Krylov Preconditioner, and Coarse Solver nodes. Use it to set up an additive-, multiplicative-, hybrid- or symmetric Schwartz overlapping domain decomposition solver. A default Domain Solver node is also added.

Domain decomposition divides the modeling domain into subdomains where the equations in the subdomains are easier to solve. The total solution is then obtained by iterating between the computed solutions for each subdomain using the currently known solutions from the other subdomains as boundary conditions. The domain decomposition solver is efficient for distributed memory systems (cluster computing) and as a more memory-efficient alternative to a direct solver for large problems.

Also see The Domain Decomposition Solver.

GENERAL

Select a **Solver**—**Multiplicative Schwarz** (the default), **Additive Schwarz**, **Hybrid Schwarz**, or **Symmetric Schwarz**.

For any **Solver**, enter values or choose an option as required:

- **Number of iterations**. The default is 1.
- **Number of subdomains**. The default is 2. The subdomain partition is created from an element partition on the solver level.
- **Maximum number of DOFs per subdomain**. The default is 100,000 DOFs. The solver tries to not create subdomains larger than this and increases the number of subdomains to fulfill the target. The lowest value accepted is 1000.
- **Maximum number of nodes per subdomain**. The default is 1. This option is only relevant in cluster computations. Each subdomain is then handled by the selected number of nodes.
- **Additional overlap**. The default is 1 mesh element. Each subdomain in the initial (non-overlapping) partition is extended via the connectivity of the stiffness matrix in a recursive algorithm; this setting controls the number of additional mesh elements—added to the necessary single mesh element—in the overlap between adjacent subdomains.
- **Overlap method**—**Matrix based** (the default) or **Mesh based**. The matrix-based overlap method considers the mesh connectivity whereas the mesh-based overlap method considers neighboring mesh elements. Select **Mesh based** if the matrix-based overlap generates too large overlapping subdomains.
- Select an option from the **Hierarchy generation method** list. Details below.
- For any **Solver**, select the **Recompute and clear subdomain data** check box if required. This is a computationally expensive option. The subdomain problems are factorized for each iteration and then cleared from memory. Only use this option if your model is too big to fit into main memory causing memory swapping to disk.
- If **Multiplicative Schwarz** or **Symmetric Schwarz** is chosen as the **Solver**, the **Use subdomain coloring** check box is selected by default to use a coloring technique that leads to more efficient computations for the multiplicative and symmetric methods because they require the global residual to be updated after each subdomain. The coloring technique gives each subdomain a color such that subdomains with the same color are disjoint and can be computed in parallel before the residual is updated. Click to clear the check box as required.

Hierarchy Generation Method

Select an option from the **Hierarchy generation method** list to specify how to generate the multigrid levels:

- **Lower element order first (any)**. The default. Generates first a number of levels by lowering the order (by one) of the used shape functions. If there are no shape functions that can be lowered, the mesh is coarsened.
- **Coarse mesh and lower order**. Combines lowering of the used shape function order and a coarsening of the mesh.
- **Lower element order first (all)**. Generates first a number of levels by lowering the order (by one) of all the used shape functions. If this is not possible the mesh is coarsened.

- **Coarse mesh**. Does not change the order.
- **Lower element order and refine (all)**. Generates a number of levels by lowering the order (by one) of all the used shape functions. If this is not possible the mesh is refined a number of times. The mesh solved for can with this method be a finer one than the one selected under the study node.
- **Lower element order and refine (any)**. Generates a number of levels by lowering the order (by one) of the used shape functions. If there are no shape functions that can be lowered, the mesh is refined. The mesh solved for can with this method be a finer one than the one selected under the Study node.
- **Refine mesh**. Does not change the order.
- **Manual**. Use this setting to select multigrid levels from the existing ones. You then specify the multigrid levels to use in the **Use multigrid levels** list and the multigrid levels to assemble the discrete differential operators on in the **Assemble on multigrid levels** list. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☒), and **Add** (+) buttons to configure the lists of multigrid levels.

For any **Hierarchy generation method** (except Manual), additional settings are available:

- In the **Use hierarchy in geometries** list, select the geometries to apply multigrid to. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☒), and **Add** (+) buttons to configure the list of geometries.
- The **Assemble on all levels** check box is selected by default to assemble the discrete differential operators on all meshes. Otherwise these operators are formed using the restriction and prolongation operators. Click to clear the check box as required.

When **Coarse mesh and lower order**, **Lower element order first (all)**, **Lower element order first (any)**, or **Coarse mesh** are selected from the **Hierarchy generation method** list:

- Select the **Keep generated multigrid levels** check box to save the meshes for all multigrid levels under the mesh node.
- Enter a **Mesh coarsening factor** to select the degree of coarsening to apply to the meshes when using mesh coarsening as the multigrid hierarchy generation method. The higher this number is the more aggressive the mesh coarsening is. The default is 2.

When **Lower element order and refine (all)**, **Lower element order and refine (any)**, or **Refine mesh** are selected from the **Hierarchy generation method** list, select a **Refinement method** to refine the multigrid levels when using mesh refinement as the multigrid hierarchy generation method. The options are:

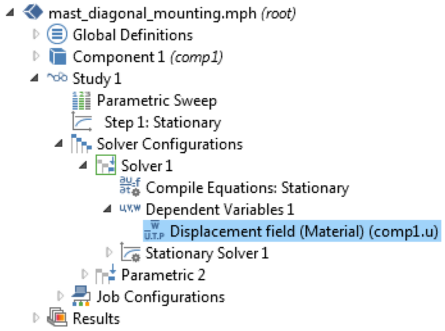
- **Split longest side**. The default method. Elements are subdivided such that the longest side in each element is always split. This yields not so many new elements, while also preserving mesh quality.
- **Regular refinement**. Elements are subdivided in a regular manner.

Domain Solver

The **Domain Solver** node (↻) is an auxiliary attribute subnode used by the Domain Decomposition attribute node. This node does not have any settings. Instead, its purpose is to administrate domain solvers for a domain-decomposition solver.

Field

The **Field** node (W U.T.P) is an attribute node that handles settings for field variables. Each field variable has a separate **Field** node. This attribute is used together with the Dependent Variables node. The Field node name matches the name of the variable.



- mast_diagonal_mounting.mph *(root)*
  - Global Definitions
  - Component 1 *(comp1)*
  - Study 1
    - Parametric Sweep
    - Step 1: Stationary
    - Solver Configurations
      - Solver 1
        - Compile Equations: Stationary
        - Dependent Variables 1
          - Displacement field (Material) (comp1.u)
        - Stationary Solver 1
      - Parametric 2
    - Job Configurations
  - Results

Figure 19-10: An example of a Field node, which takes the name of the variable, in this case Displacement field (Material).

GENERAL

The **Field components** section displays the variable names for the field's components. Also, when internal extra variables are used, these are displayed here as **Internal variables**.

The **Solve for this field** check box is available if the Dependent Variables **Defined by study step** setting is **User defined**. It controls whether to solve for the field (variable) or not. If the variable is not solved for its value is determined by the settings in the **Values of Variables Not Solved For** section of the parent **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not. A variable can still be solved for despite not being stored in output and vice versa.

SCALING

Select a **Method** to control the scaling of a variable.

> Unless **From parent** is selected, specifying a **Method** for a variable here overrides the **Method** selected in the **Scaling** section of the corresponding **Dependent Variables** operation node.

Select:

- **Automatic** to get an automatically determined scaling.
- **From parent** to use the scaling type selected in the **Method** list in the **Scaling** section of the corresponding **Dependent Variables** operation node.
- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling. If **Manual** is chosen, enter a value in the **Scale** field
- **None** to get no scaling.

Fully Coupled

The **Fully Coupled** attribute node ( ) uses a damped version of Newton's method, or for stationary problems, a double dogleg method, to handle parameters for a fully coupled solution approach. It can be used with the

Stationary Solver and the Time-Dependent Solver.

An alternative to the fully coupled approach is given by the segregated solver, which you control with the Segregated node. Although several **Fully Coupled** and **Segregated** attribute subnodes can be attached to an operation node, only one can be active at any given time.

For more information about the settings below, see:

- The Fully Coupled Attribute and the Double Dogleg Method
- Damped Newton Methods
- Termination Criterion for the Fully Coupled and Segregated Attribute Nodes

GENERAL

Select a **Linear solver** for linear systems that appear in the corresponding solver configuration. The available solvers are attribute nodes of the types Direct and Iterative.

METHOD AND TERMINATION

Select a **Nonlinear method** to control which damping factor to use in the damped Newton iterations. Select:

- **Automatic (Newton)** to let the solver automatically determine a damping factor in each iteration of Newton's method. Go to Automatic (Newton) and Automatic Highly Nonlinear (Newton) for settings.
- **Constant (Newton)** to manually specify a constant damping factor that is used in all iterations of Newton's method. Go to Constant (Newton) for settings.
- **Automatic highly nonlinear (Newton)** if the solver does not converge with **Automatic (Newton)** first. It is similar to **Automatic (Newton)** but this method can make the solver more careful when solving highly nonlinear problems. Go to Automatic (Newton) and Automatic Highly Nonlinear (Newton) for settings.
- For stationary problems, choose **Double dogleg** to use the double dogleg nonlinear solver. Go to Double Dogleg for settings.

Automatic (Newton) and Automatic Highly Nonlinear (Newton)

If **Automatic (Newton)** or **Automatic highly nonlinear (Newton)** is chosen, enter values or expressions for:

- **Initial damping factor**, to specify a damping factor for the first Newton iteration. The default value is $1.0 \cdot 10^{-4}$.
- **Minimum damping factor**, to specify the smallest allowed damping factor. The default value is $1.0 \cdot 10^{-8}$.
- **Restriction for step-size update**, to specify a factor that limits how much the damping factor is allowed to change in a Newton iteration. The damping factor can change up or down by at most this factor. The default is 10.

Select an option from the **Use recovery damping factor** list—**Automatic** (the default), **On**, or **Off**.

The default, **Automatic**, is equivalent to **On** for stationary problems and **Off** for time-dependent problems. For stationary parametric continuation problems, **Automatic** corresponds to **On** when solving for the first parameter value and **Off** when solving for subsequent parameter values.

- Choose **Off** if a damping factor smaller than the **Minimum damping factor** is required. The nonlinear solver then terminates.
- If **On** is selected, the nonlinear solver takes a Newton step using the constant damping factor, which is defined in the **Recovery damping factor** field.
- The default **Recovery damping factor** is 0.75 when **Automatic** or **On** is selected.
- Continue with the Termination Technique and Termination Criterion settings that follow.

Constant (Newton)

If **Constant (Newton)** is chosen:

- Enter a value for the **Damping factor** to specify a constant damping factor for Newton's method. The default is 1.
- With a Time-Dependent Solver select the **Limit on nonlinear convergence rate** check box to force the nonlinear solver to terminate as soon as the convergence is estimated to be too slow. The default is 0.9. Enter a limit on the convergence rate in the field as required.
- Choose a **Jacobian update**—**Minimal** (the default), **On every iteration**, and **Once per time step**.
  - **On every iteration** computes a new Jacobian for all iterations of Newton's method.
  - With a Stationary Solver, **Minimal** and **On first iteration** are identical. A new Jacobian is computed for the first iteration of Newton's method only.
  - With a Time-Dependent Solver, **On first iteration** computes a new Jacobian for each new nonlinear system of equations and **Minimal** reuses the Jacobian for several nonlinear systems whenever deemed possible.
- Continue with the Termination Technique—Constant (Newton), Pseudo Time Stepping—Constant (Newton), and Termination Criterion settings that follow.

Double Dogleg

If **Double dogleg** is chosen for stationary problems:

- Enter a value for the **Initial damping factor**, to specify a damping factor for the first Newton iteration. The default value is $1.0 \cdot 10^{-4}$.
- Choose a **Residual scaling**—**Field-wise** or **Uniform. Field-wise** scales the equations based on the field-wise sizes of the initial residual. If **Uniform** is selected, the algorithm terminates on the relative residual based on the initial residual.
- Continue with the Termination Technique settings that follow.

Termination Technique

For any **Nonlinear method**, select a **Termination technique** to control how the Newton iterations are terminated. Select:

- **Tolerance** to terminate the Newton iterations when the estimated relative error is smaller than a specified tolerance. Then enter the **Maximum number of iterations** to limit the number of Newton iterations. When the maximum number of iterations have been performed Newton's method is terminated even if the tolerance is not fulfilled.
- **Iterations or tolerance** to terminate the Newton iterations when the estimated tolerance is smaller that a specified tolerance or after a specified number of iterations, whichever comes first. Then enter the **Number of iterations** to specify a fixed number of iterations to perform.
- If **Tolerance** or **Iterations or tolerance** are set as the **Termination technique**, then enter a **Tolerance factor** to modify the tolerance used for termination of the Newton iterations. The actual tolerance used is this factor times the value specified in the **Relative tolerance** field in the **General** sections of the Stationary Solver and Time-Dependent Solver.

Termination Technique—Constant (Newton)

When **Constant (Newton)** is the Nonlinear method, choose **Iterations** to terminate the Newton iterations after a fixed number of iterations. Enter the **Number of iterations** to specify a fixed number of iterations to perform.

Pseudo Time Stepping—Constant (Newton)

When **Constant (Newton)** is the **Nonlinear method**, and for a Stationary Solver, turn **Pseudo time stepping** on or off. See Pseudo Time Stepping for more information.

When set **On**, specify the regulator parameters:

- **Initial CFL number**. The default is 5.
- **PID regulator-Proportional**. The default is 0.65.
- **PID regulator-Derivative**. The default is 0.05.
- **PID regulator-Integrative**. The default is 0.05.
- **Target error estimate**. The default is 0.1.

Termination Criterion

This section is not available if the termination technique is set to **Iterations**.

For a Stationary Solver select a **Termination criterion** to control how the Newton iterations are terminated. Select:

- **Solution** to terminate the Newton iterations on a solution-based estimated relative error.
- **Residual** to terminate the Newton iterations on a residual-based estimated relative error.
- **Solution or residual** to terminate the Newton iterations on the minimum of the solution-based and residual-based estimated relative errors. Enter a scalar **Residual factor** multiplying the residual error estimate. The default is 1000.

RESULTS WHILE SOLVING

See Results While Solving in the Common Study Step Settings section. Also see Getting Results While Solving.

For an example using a Stationary Solver, see Tubular Reactor: model library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

For an example using a Time-Dependent Solver, see Sloshing Tank: model library path **COMSOL_Multiphysics/Fluid_Dynamics/sloshing_tank**.

Incomplete LU

The **Incomplete LU** node ( ) is an attribute that handles parameters for linear system solvers/preconditioners that use incomplete LU factorization. Right-click an Iterative, Krylov Preconditioner, Presmoother, Postsmoother, or Coarse Solver attribute node to add an **Incomplete LU** node. Also see About Incomplete LU.

GENERAL

In this section you specify the properties of the incomplete LU preconditioner.

Select a **Solver**. Select:

- **Incomplete LU** (the default) to use a solver designed specifically for incomplete LU factorization.
- **SPOOLES** to use the sparse object-oriented linear equations solver SPOOLES.

Drop Using

For **Incomplete LU**, select an option from the **Drop using** list to specify a drop rule. See Selecting a Drop Rule. Select:

- **Tolerance** (the default) to let the solver drop (neglect) elements that have small enough absolute values. Tune the sizes of the neglected elements either in the **Drop tolerance** field or using the accompanying slide bar. A larger tolerance neglects more elements.
- **Fill ratio** to let the solver keep a certain fraction of the elements. The elements with largest absolute values are kept. Tune the number of nonzero elements in the incomplete factorization using either the **Fill ratio** field or the accompanying field. A smaller fill ratio neglects more elements.

Drop Tolerance

For both **Incomplete LU** and **SPOOLES**, use the **Drop tolerance** field or the accompanying slide bar to tune the maximum allowed sizes of dropped (neglected) elements (default: 0.01). A smaller drop tolerance means that the preconditioner drops fewer elements and so the preconditioner becomes more accurate. This leads to fewer iterations in the iterative solver, but memory requirements and preconditioning time increase. A larger drop tolerance means that the preconditioner drops more elements and so memory use and preconditioning time decrease. In this case, however, the preconditioner becomes less accurate, which leads to more iterations in the iterative solver, or, if the drop tolerance is too high, to no convergence at all. Often it is most efficient to use as high a drop tolerance as possible; that is, choose it so that the iterative solver barely converges.

Respect Pattern

For **Incomplete LU**, by default the solver never drops elements in positions where the original matrix is nonzero. Clear the **Respect pattern** check box to allow the solver to drop also such elements.

Pivot Threshold

For both **Incomplete LU** and **SPOOLES**, use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold (default: 1). This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column. The solver permutes rows for stability. In any given column, if the absolute value of the diagonal element is less than the pivot threshold times the largest absolute value in the column, it permutes rows such that the largest element is on the diagonal. Thus the default value 1 means that it uses partial pivoting.

Number of Iterations and Relaxation Factor

For **Incomplete LU**—once the approximate factors L and U have been computed—you can use the incomplete LU factorization as an iterative preconditioner/smoother. Here, $M = (LU)/\omega$, where ω is a relaxation factor, and L and U are the approximate factors. Use the **Number of iterations** field to specify how many iterations to perform (default: 1). The relaxation factor ω is similar to the one used by, for example, the **SOR** node. Specify such a factor in the **Relaxation factor** field (default: 1).

Preordering Algorithm

For **SPOOLES**, use the **Preordering algorithm** list to select one of the following preorderings:

- **Nested dissection** (NS, the default)
- **Minimum degree**
- **Multisection** (MS)
- **Best of ND and MS** (the best of nested dissection and multisection)

Iterative

The **Iterative** node (▨) is an attribute that handles settings for iterative linear system solvers. You can use it with an Eigenvalue Solver, Stationary Solver, or Time-Dependent Solver, for example.

An alternative to the iterative linear system solvers is given by direct linear system solvers, which are handled via the Direct attribute node. Although several attribute subnodes for solving linear systems can be attached to an operation node, only one can be active at any given time.

The iterative solver iterates until a relative tolerance is fulfilled. You specify this tolerance in the **Relative error** field of the operation node that this attribute belongs to.

Also see The Iterative Solvers, Iterative Solver Types, and Selecting a Preconditioner for an Iterative Linear System Solver for more details about the settings under **General**.

GENERAL

Use the **Solver** list to select an iterative linear system solver. Select:

- **GMRES** to use the restarted GMRES (Generalized Minimum RESidual) iterative method.
- **FGMRES** to use the restarted FGMRES (Flexible Generalized Minimum RESidual) iterative method.
- **BiCGStab** to use the BiCGStab (BiConjugate Gradient Stabilized) iterative method.
- **Conjugate gradients** to use the Conjugate gradients iterative method.
- **Use preconditioner** to use the currently active preconditioner attached to this **Iterative** node as the linear system solver. This solver uses the settings for **Factor in error estimate** and **Maximum number of iterations** from the **Error** section of the corresponding **Iterative** node.

If **GMRES** or **FGMRES** is selected, specify the **Number of iterations before restart** that the solver performs until it restarts (the default is 50). There is no guarantee that a restarted GMRES converges for a small restart value. A larger restart value increases the robustness of the interactive procedure, but it also increases memory use and computational time. For large problems, the computational cost to produce a preconditioner of such quality that the termination criteria are fulfilled for a small number of iterations and for a small restart value is often large. For those problems it is often advantageous to set up a preconditioner with a somewhat lesser quality and instead increase the restart value or iterate more steps. Doing so typically increases the condition number for the preconditioned system, so an increase in the error-estimate factor might be needed as well.

If **FGMRES** is selected, specify whether you are **Preconditioning** the linear system matrix from the **Left** or from the **Right**. Normally, this setting does not significantly influence the convergence behavior of the selected solver. The default choice is left preconditioning. Normally, the two versions of GMRES have similar convergence behavior (see Ref. 11). If the preconditioner is ill-conditioned there could, however, be differences in the behavior. For the conjugate gradient method this choice only affects the convergence criterion and not the algorithm itself.

ERROR

The stopping criteria for the iterative solvers are based on an error estimate, which checks if the relative residual times a stability constant ρ is less than a tolerance. This tolerance is specified in the **Relative error** field of the operation node that this attribute belongs to.

Use the **Factor in error estimate** field to set ρ, which serves as a safety factor to avoid premature termination of the iterations due to, for example, ill-conditioning of the matrix A or poor preconditioning (default: 400). A larger value of ρ increases the number of iterations, but decrease the chance that the iteration is terminated too early. To avoid false termination, the main solver continues iterating until the error check passes or until the step size is smaller than about $2.2\cdot10^{-14}$.

For information about the convergence criteria used by the iterative solvers and the **Relative tolerance** and **Factor in error estimate** fields, see Convergence Criteria for Iterative Solvers.

Use the **Maximum number of iterations** field to enter a maximum number of iterations that the iterative solver is allowed to take (default: 10,000 iterations). When this number of iterations has been performed without reaching the specified tolerance the solver stops with an error message.

By choosing the appropriate option from the **Validate error estimate** list the error estimate for left preconditioned solvers can be validated. **No** implies no error estimate validation, whereas **Automatic** or **Yes** does. By default the validation is enabled with the **Automatic option** meaning that it is performed but preconditioner warnings are only issued if the iterative solver fails with an error. To avoid false termination, the main solver continues iterating until the error check passes or until the step size is smaller than about $2.2 \cdot 10^{-14}$. The **Validate error estimate** setting is propagated recursively and applies to all children with left preconditioning.

For an example using a Stationary Solver, see Micromixer: model library path **COMSOL_Multiphysics/Fluid_Dynamics/micromixer**.

Jacobi

The **Jacobi** node ( ) is an attribute that handles settings for the Jacobi (or diagonal scaling) method. Right-click the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver attribute nodes to add a **Jacobi** feature.

The Jacobi method provides a simple and memory-efficient solver/preconditioner/smoother based on classical iteration methods for solving a linear system of the form $Ax = b$. Given a relaxation factor ω (usually between 0 and 2), a sweep of the Jacobi (diagonal scaling) method transforms an initial guess x_0 to an improved approximation $x_1 = x_0 + M^{-1}(b - Ax_0)$, where $M = D/\omega$, and D is the diagonal part of A.

GENERAL

Settings When Not Used With Coarse Solver

Enter the **Number of iterations** to perform when this node is used as a preconditioner or smoother. This setting is not considered when the attribute is used as a linear system solver (with the **Use preconditioner** option in the **Solver** list of the **Iterative** attribute node). The solver then iterates until the relative tolerance specified by the corresponding operation node is fulfilled rather than performing a fixed number of iterations.

Enter a **Relaxation factor** to specify a scalar relaxation factor ω. The allowed values of this factor are between 0 and 2. The default is 1.

If you have the Molecular Flow Module plus the Particle Tracing Module, see Molecular Flow Through an RF Coupler: model library path **Molecular_Flow_Module/Benchmark_Models/rf_coupler**.

Settings When Used with a Coarse Solver

If this node is used with a Coarse Solver, select a **Termination technique** to determine how to terminate the solver. Select **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used, or **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.

If **Fixed number of iterations** is selected enter a value for each of the following:

- **Number of iterations** to perform. The default is 10.
- **Relaxation factor** to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. The default is 1.

If **Use tolerance** is selected, enter a value for each of the following:

- **Relative tolerance** to specify the termination tolerance. The default is 0.1.
- **Maximum number of iterations** that the solver is allowed to take. When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver is automatically stopped with an error message.
- **Relaxation factor** to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. The default is 1.

Krylov Preconditioner

The **Krylov Preconditioner** node ( ) is an attribute that handles settings for Krylov-type methods. The settings are similar to the Iterative attribute node; the difference is that this node is an auxiliary attribute node, whereas Iterative handles settings for a main iterative solver.

Right-click the Iterative, Presmoother, Postsmoother, and Coarse Solver attribute nodes to add a **Krylov Preconditioner**.

GENERAL

Select a linear system **Solver**—**GMRES** (the default), **FGMRES**, **BiCGStab**, or **Conjugate gradients**. See Iterative Solver Types for descriptions.

- When **GMRES** or **FGMRES** is selected, enter a **Number of iterations before restart** to specify how many iterations the solver should take between each restart. A larger number increases robustness but also memory use and computational time.
- When **GMRES**, **BiCGStab**, or **Conjugate gradients** is selected, select an option from the **Preconditioning** list to specify whether to precondition the linear system matrix from the **Left** or from the **Right**. Normally, this setting does not significantly influence the convergence behavior of the selected solver.
- For all **Solver** types, select a **Termination technique**:
  - **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the solver is used.
  - **Use tolerance** to terminate the solver when a tolerance is fulfilled. Then enter a **Relative tolerance** (default: 0.1) and the **Maximum number of iterations** the solver is allowed to take. When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field, the solver is automatically stopped with an error message.
  - **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first. Then enter a **Relative tolerance** (default: 0.1).
- For all **Solver** types, enter the **Number of iterations**. The default is 10 or, when a tolerance is used, 500.

ERROR

Select an option from the **Validate error estimate** list—**Automatic** (the default), **Yes**, or **No**. By choosing the appropriate option from the **Validate error estimate** list the error estimate for left preconditioned solvers can be validated. **No** implies no error estimate validation, whereas **Automatic** or **Yes** does. By default the validation is enabled with the **Automatic option** meaning that it is performed but preconditioner warnings are only issued if the iterative solver fails with an error. This setting is propagated recursively and applies to all children with left preconditioning.

Lower Limit

The **Lower Limit** node ( ) makes it possible to impose restrictions on degrees of freedom (field variables). These restrictions are checked after each substep of a segregated approach. The restriction is performed without any regards to the equations, so use this restriction with care. Enforcing a lower limit on a field variable can be useful to ensure positivity of a volume fraction of particles or the positivity of turbulence model variables, for example. Right-click a Segregated node to add a **Lower Limit** node.

LOWER LIMIT

By default, no variables have active lower limits. To activate lower limits for field variables, use the **Lower limits (field variable)** field to specify the variables and their scalar lower limits as space-separated pairs: `field_variable_1 limit_value_1 field_variable_2 limit_value_2`, and so on. For example, to impose a lower limit of 0.25 for the value of a field **u** in Component 1, enter `comp1.u 0.25`.

Lumped Step

The **Lumped Step** node ( ) is available when using the Segregated attribute node. This step is intended for speeding up the computation of any L_2-projections, stemming from the identity operator, appearing as single physics within a multiphysics problem. Using the lumped step approximates the Jacobian matrix resulting from finite element discretization when solving the linear system for the unknown field variables. The approximation consists of replacing the Jacobian by a diagonal matrix with row sums. In doing so, take care to ensure that the resulting approximate Jacobian is invertible (nonsingular).

Multigrid

The **Multigrid** solver ( ) is used to set up a geometric multigrid (GMG) solver or an algebraic multigrid (AMG) solver. Right-click the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver attribute node to add a **Multigrid** solver.

Linear in the *COMSOL API Reference Manual*

GENERAL

Select a **Solver**—**Geometric multigrid** or **Algebraic multigrid**. For either choice, enter the:

- **Number of iterations**. The default is 2.
- Select a **Multigrid cycle**—**V-cycle** (the default), **W-cycle**, or **F-cycle**. If **Multigrid cycle** is chosen, the settings are the same as for the geometric multigrid (GMG) and algebraic multigrid (AMG) solvers.
- Enter the **Number of multigrid levels** to generate (the default is 1 for **Geometric multigrid** and 5 for **Algebraic multigrid**).

Geometric Multigrid

If **Geometric multigrid** is chosen, see The Geometric Multigrid Solver/Preconditioner for more information about this option.

The Hierarchy Generation Method options are described for the Direct node.

If **None** is selected, no coarse mesh is used in addition to the fine mesh. This can lead to severe reduction in convergence rate but saves memory.

Algebraic Multigrid

If **Algebraic multigrid** is chosen, see The Algebraic Multigrid Solver/Preconditioner for more information about this option. In addition to the settings above, the following settings control the automatic construction of the multigrid hierarchy:

- Enter a **Maximum number of DOFs at coarsest level**. The default is 5000. Coarse levels are added until the number of DOFs at the coarsest level is less than the max DOFs at coarsest level or until it has reached the number of multigrid levels.
- Enter a value or use the slider to set the **Quality of multigrid hierarchy**. Higher quality means faster convergence at the expense of a more time consuming set-up phase. For instance, if the linear solver does not converge or if it uses too many iterations, try a higher value to increase the accuracy in each iteration, meaning fewer iterations.

If the algebraic multigrid algorithm runs into memory problems, try a lower value to use less memory. The range goes from 1 to 10 where 10 gives the best quality. The default is 3.

- The Multigrid Solvers
- Multigrid Level

Parametric

The **Parametric** node () is an attribute node that handles settings for parameter stepping using a parametric solver. This node can be used together with a Stationary Solver node.

There is also an option to run a Stationary study with an **Auxiliary sweep**, with or without a continuation parameter. When a continuation parameter is selected the continuation algorithm is run, which assumes that the sought solution is continuous in these parameters. If no continuation parameter is given, a plain sweep is performed where a solution is sought for each value of the parameters. In both cases, a **Stationary Solver** node plus a **Parametric** attribute is used. The parametric solver is the algorithm that is run when a Parametric attribute node is active under a Stationary Solver. Similarly the adaptive solver is the algorithm that is run when an Adaptive Mesh Refinement node is active under a Stationary Solver.

In order to run a parametric continuation, select the **Auxiliary sweep** check box under **Study Extensions** for the Stationary or Frequency Domain study step. Then on the study settings window, define the parameters in the table and choose one from the **Run continuation for** list.

- The Adaptive and Parametric Solver Logs
- The Parametric Solver Algorithm

GENERAL

Select an option from the **Defined by study step** list to specify if the settings are synchronized with the Stationary or Frequency Domain study step, in which case this section does not require any input. The **Run continuation for** list also displays the same settings made under **Study Extensions**.

The continuation algorithm is used if a parameter is selected. Normally the step size, damping, and predictor settings are automatic. If required, you can edit the settings in the **Continuation** section described next.

To edit the settings, select **User defined** to modify the sweep type, parameter table, reuse solution for previous step setting, and the parameter to run continuation for. These settings are the same as described in Common Study Step Settings under **Study Extensions**.

Exactly how the parameter values are used by the solver is determined by the **Sweep type** and the option **Parameters to store** in the **Output** section as described below. If more than one parameter name is specified, the lists of parameter values are interpreted as follows. Assume that the parameter names are `p1` and `p2`, and that `p1` has the list 1 3 and `p2` has the list 2 4:

- If **Specified combinations** is chosen, the solver first uses `p1` equal to `1` and `p2` equal to `2`. Thereafter, it uses `p1` equal to `3` and `p2` equal to `4`.
- If **All combinations** is chosen, the solver uses this order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

To determine what the solver does when there is a solver error or when the continuation backtracking fails, select an option from the **On error** list. Select

- **Stop** (the default) to stop the parametric sweep and only return solutions before the error.
- **Store empty solution** to continue the parametric sweep and store an empty (NaN) solution for this step (or for the remaining continuation).

Using **Store empty solution** can be useful if you need to sweep over many different combinations of parameters and it is unknown which one will solve. It can also be useful when doing frequency sweeps where frequencies close to resonances fail.

CONTINUATION

By default the solver selects the parameter steps automatically based on the values entered in the **Parameter values** field in the **General** section.

Click to select the **Tuning of step size** check box to edit these settings:

- **Initial step size** field to enter a positive number that determines the magnitude of the first parameter step.
- **Minimum step size** field to specify a safeguard against too small parameter steps.
- **Maximum step size** field to specify an upper bound on the parameter step size. Use this if you suspect that the solver tries to take unnecessarily long steps.

Use the **Use initial damping factor for all parameter steps** list to control the initial damping factor for the nonlinear solvers for the parameter steps.

- Select **On** to use the given **Initial damping factor** for the nonlinear solvers for all parameter steps.
- Select **Off** to use the initial damping factor only for the first parameter step.
- Select **Automatic** (default) to use the initial damping factor only for the first parameter step when the **Automatic (Newton)** or **Automatic highly nonlinear (Newton)** nonlinear solver method is used in the Fully Coupled solver node and use the initial damping factor in all steps for other solver combinations.

Use the **Predictor** list to control how the initial value for the next parameter value is determined. Select:

- **Automatic** (the default) to let the parametric solver choose a constant or linear predictor based on the type of stationary solver (a constant predictor for segregated solvers and a linear predictor for fully coupled solvers).

- **Constant** to use the solution for the present parameter value as initial guess.
- **Linear** to compute the initial guess by following the tangent to the solution curve at the present parameter value.

This option is overridden and **Constant** used instead if you are solving for more than one parameter (that is, when you have entered more than one parameter name in the **Parameter names** field).

LOAD CASE

This section displays the settings made under **Study Extensions** for the Stationary study step; it is synchronized with the study settings.

OUTPUT

Use the **Parameters to store** list to control at what parameter values the solver stores a solution. Select:

- **Steps given** to store solutions at the parameter values entered in the **Parameter values** field in the **General** section.
- **Steps taken by solver** to store solutions at all parameter values where the solver has computed a solution. This option can generate solutions in-between the values specified by the **Parameter values** field in the **General** section if the solver needs to take shorter steps than specified by the values in that field.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk instead of in the computer's internal memory.

RESULTS WHILE SOLVING

See Results While Solving in the Common Study Step Settings section. Also see Getting Results While Solving.

CLUSTER SETTINGS

Select the **Distribute parameters** check box to distribute the parameters on several computational nodes. If the problem is too large to run on a single node you can enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a nondistributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

Buoyancy Flow in Free Fluids: model library path
COMSOL_Multiphysics/Fluid_Dynamics/buoyancy_free

Postsmoother

The **Postsmoother** node () is an auxiliary attribute node used by the Multigrid attribute node. This attribute does not have any settings. Instead, its purpose is to administrate postsmoothers for a multigrid solver.

Presmoother

The **Presmoother** node () is an auxiliary attribute node used by the Multigrid attribute node. This attribute does not have any settings. Instead, its purpose is to administrate presmoothers for a multigrid solver.

Previous Solution

The **Previous Solution** node ( ) is an optional attribute node of the Parametric attribute node. It handles field variables that have to be accessed at a previous parameter value or time.

Use the **Variables** list to specify which variables to associate with the previous parameter value rather than the present one.

Use the **Linear solver** list to select a solver for the linear systems associated with the quantities specified by **Variables**. The available solvers are attribute node of the types Direct and Iterative.

SCGS

The **SCGS** node ( ) is an attribute that handles the SCGS (symmetrically coupled Gauss-Seidel) solver, which is useful as a preconditioner for solving the Navier-Stokes equations and similar fluid flow problems. See The SCGS Solver for more information.

The settings window includes the following sections:

MAIN

Settings When Not Used With Coarse Solver

If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute node is being used as a preconditioner or smoother (default: 2).

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the **Coarse Solver** is used.
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

Settings When Used With Any Attribute Node

Use the **Relaxation factor** field to specify a scalar relaxation factor ω. The allowed values of this factor are between 0 and 2 (default: 0.8).

Use the **Block solver** list to specify how to solve the Vanka block linear systems by selecting one of these options:

- **Direct, stored factorization** (the default) to store the factorization. If two SCGS smoothers are used as presmoother and postsmoother of a Multigrid solver, with similar enough settings, they share the same stored factorization, which means that they only use half the memory.

- **Direct** to use a direct solver. The direct solver is slower than the default option to store the factorization, but it uses less memory.
- **Iterative, GMRES** to use the iterative method GMRES.

If you use the SCGS algorithm as preconditioner, or as a smoother to a multigrid preconditioner when either of GMRES, Conjugate gradients, or BiCGStab is used as the linear system solver, use the **Direct, stored factorization** or the **Direct** option in the **Block solver** list in order to get a stationary preconditioner.

The **Iterative, GMRES** option in the **Block solver** list can be useful if you use the FGMRES method as linear system solver because it can handle preconditioners that are not stationary. The GMRES option can also be useful if you use the SCGS algorithm as a smoother to a multigrid solver because GMRES can in some cases be faster than the direct solver if set to a high tolerance, although this advantage is less pronounced with SCGS than Vanka due to the smaller block size used by SCGS.

When **GMRES** has been selected in the **Block solver** list the following options become available. Use the **Tolerance** field to specify the termination tolerance of GMRES (default: 0.02). Use the **Number of iterations before restart** field to specify how many iterations the solver should take between each restart (default: 100).

From the **Method** list, select one of the following methods (see above):

- **Mesh element lines and vertices** (the default)
- **Mesh elements**
- **Mesh element lines**

Select the Vanka check box and then use the **Variables** list to specify variables to include in a Vanka block approach.

Select the **Blocked version** check box (selected by default) to use a version of the SCGS method that is optimized for parallel computations.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations (default: 1) to perform for degrees of freedom not involved in the SCGS blocks.

Use the **Relaxation factor** field to specify a scalar relaxation factor for the iterations specified in the **Number of secondary iterations** field (default: 1). The allowed values of this factor are between 0 and 2 (default: 0.5).

Segregated

The **Segregated** node ( ) is an attribute that handles parameters for a segregated solution approach. This attribute makes it possible to split the solution process into substeps. Each substep uses a damped version of Newton’s method.

The attribute can be used together with the Stationary Solver and Time-Dependent Solver nodes. An alternative to the segregated approach is given by the coupled solver, which is handled with the Fully Coupled attribute node. Although several **Fully Coupled** and **Segregated** attribute nodes can be attached to an operation node, only one can be active at any given time.

To add substeps to a segregated iteration, right-click the **Segregated** node. One segregated iteration consists of executing each active Segregated Step in the order shown in the model tree.

> The convergence properties of a model might depend on the order of the segregated steps. You can move the Segregated Step nodes to change the order in which the solver runs each step.

For more information about the settings below, see:

- The Segregated Solver
- Damped Newton Methods
- Pseudo Time Stepping
- Termination Criterion for the Fully Coupled and Segregated Attribute Nodes

GENERAL

Select a **Termination technique** to control how the segregated iterations are terminated. Select:

- **Tolerance** (the default) to terminate the segregated iterations when the estimated relative error is smaller than a specified tolerance.
- **Iterations or tolerance** to terminate the segregated iterations when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.
- **Iterations** to terminate the segregated iterations after a fixed number of iterations.

Then based on the **Termination technique** selected:

- If **Tolerance** is selected, enter a **Maximum number of iterations** to limit the number of segregated iterations (default: 10). When the maximum number of iterations have been performed the segregated method is terminated even if the tolerance is not fulfilled.
- If **Tolerance** or **Iterations or tolerance** is selected, enter a **Tolerance factor** to modify the tolerance used for termination of the segregated iterations. The actual tolerance used is this factor times the value specified in the **Relative tolerance** field in the **General** sections of the **Stationary Solver** and **Time-Dependent Solver** nodes.
- If **Tolerance** or **Iterations or tolerance** is selected, choose a **Termination criterion** to control how the Newton iterations are terminated for stationary problems. Select:
  - **Solution** to terminate the Newton iterations on a solution-based estimated relative error.
  - **Residual** to terminate the Newton iterations on a residual-based estimated relative error.
  - **Solution or residual** to terminate the Newton iterations on the minimum of the solution-based and residual-based estimated relative errors. Then enter a scalar **Residual factor** (default: 1000) that multiplies the residual error estimate.
- If **Iterations** or **Iterations or tolerance** is selected, enter a **Number of iterations** to specify a fixed number of iterations to perform. The default is 1.

With a Time-Dependent Solver, also select the **Limit on nonlinear convergence rate** check box to force the nonlinear solver to terminate as soon as the convergence is estimated to be too slow. Enter a limit on the convergence rate in the accompanying field.

For a Stationary Solver, select **On** or **Off** from the **Pseudo time-stepping** list and then if enabled, specify these regulator parameters:

- **Initial CFL number**
- **PID regulator-Proportional**

- **PID regulator-Derivative**
- **PID regulator-Integrative**
- **Target error estimate**

RESULTS WHILE SOLVING

See Results While Solving in the Common Study Step Settings section. Also see Getting Results While Solving.

Segregated Step

The **Segregated Step** node ( ) handles settings for one substep of a segregated iteration. This attribute uses a damped version of Newton's method and can be used together with a Segregated attribute node.

For more background information about the method and termination settings, see The Segregated Solver and Damped Newton Methods.

GENERAL

Use the **Variables** list to specify variables to be solved for in this segregated step.

Select a **Linear solver** for the linear systems associated with the quantities specified by **Variables**. The available solvers are attribute nodes of the types Direct and Iterative.

METHOD AND TERMINATION

See the Fully Coupled **Method and Termination** section for all settings except for the following, which has a slightly different behavior as described:

If **Constant (Newton)** is chosen as the **Nonlinear method** choose a **Jacobian update—Minimal** (the default), **On every iteration**, and **Once per time step**.

- **On every iteration** computes a new Jacobian for all iterations of Newton's method.
- **Minimal** updates the Jacobian at least once and then only when the nonlinear solver fails during time or parameter stepping. It reuses the Jacobian for several nonlinear systems whenever deemed possible.
- **On first iteration** updates the Jacobian for the first subiteration for this segregated step. When used with a Time-Dependent Solver node, the option **Once per time step** is available and the Jacobian is updated at least once per time step.

The pseudo time stepping settings are not applicable for this node.

Sensitivity

A **Sensitivity** solver node ( ) solves a sensitivity analysis problem set up in The Sensitivity Interface.

Also see The Sensitivity Analysis Algorithm. Sensitivity analysis for time-dependent problems is available with the Optimization Module.

GENERAL

In the **Objective** list, you specify the objective to use for the sensitivity analysis.

In the **Sensitivity method** list you can choose between the following options:

- **Adjoint**—The adjoint method solves for the derivatives of a single scalar objective function with respect to any number of sensitivity variables.
- **Forward**—The forward sensitivity method solves for the derivatives of all dependent variables and an optional scalar objective function with respect to a small number of sensitivity variables.

SOR

The **SOR** node ( ) handles settings for the SOR (successive over-relaxation) iterative method. Right-click the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver attribute nodes to add an **SOR** node.

See The SOR Method for more detailed information about this feature.

GENERAL

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR algorithm to use. Select:

- **SSOR** (the default) to use the symmetric SOR algorithm, which in each iteration performs one **SOR** sweep followed by one **SORU** sweep.
- **SOR** to use the forward SOR algorithm.
- **SORU** to use the backward SOR (SORU) algorithm.

Specify a scalar **Relaxation factor** ω. The allowed values of this factor are between 0 and 2 (default: 1).

The **Blocked version** check box is selected by default and it uses a blocked version of the SOR method that is optimized for parallel computations. M is then constructed from a column-permuted version of A.

Settings With and Without a Coarse Solver

- If used with a Coarse Solver, select a **Termination technique**. These are described for the SCGS attribute under Settings When Used With Coarse Solver.
- If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2).

This setting is not considered when the attribute is used as a linear system solver (with the **Use preconditioner** option in the **Solver** list of the Iterative node). The solver then iterates until it has established convergence or has reached the maximal number of iterations as specified by the corresponding Iterative node rather than performs a fixed number of iterations.

SOR Gauge

The **SOR Gauge** node ( ) handles settings for the SOR gauge iterative method. This is a method of SOR-type with added functionality useful as preconditioner/smoother for, for example, 3D magnetostatics in the AC/DC Module discretized with vector elements. In short, the added functionality consists of divergence cleaning for degrees of freedom discretized with vector elements. This node can be used together with the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver nodes.

Also see The SSOR Gauge, SOR Gauge, and SORU Gauge Algorithms.

MAIN

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR Gauge algorithm to use. Each variant first performs one ordinary SOR iteration followed by one or several divergence cleaning iterations. Select:

- **SSOR gauge** (the default) to perform an ordinary **SSOR** iteration followed by divergence cleaning.
- **SOR gauge** to perform an ordinary **SOR** iteration followed by divergence cleaning.
- **SORU gauge** to perform an ordinary **SORU** iteration followed by divergence cleaning.

Specify a scalar **Relaxation factor** ω. The allowed values of this factor are between 0 and 2 (default: 1).

The **Blocked version** check box is selected by default and it uses a version of the SOR Gauge method that is optimized for parallel computations.

Use the **Variables** list to specify variables to include in the divergence cleaning phase of an SOR Gauge iteration. By default, all vector degrees of freedom are included.

Settings With and Without a Coarse Solver

- If used with a Coarse Solver, select a **Termination technique**. These are described for the SCGS attribute under Settings When Used With Coarse Solver.
- If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2). In addition to the initial divergence cleaning, the method performs a number of cleaning iterations in each linear solver iteration. Control the number of such divergence cleaning iterations in the **Number of secondary iterations** field.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of divergence cleaning iterations to perform for each main iteration (default: 1).

SOR Line

The **SOR Line** node ( ) handles settings for the SOR line iterative method. This is a method of SOR type with added functionality useful for, for example, anisotropic meshes. It is a block SOR solver, where the blocks are formed from lines of nodes that are relatively close to each other. In addition, ordinary SSOR iterations are performed for all degrees of freedom after the SOR Line iterations have been performed. This node can be used together with the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver nodes. Also see The SOR Line Algorithm.

MAIN

Settings When Used With Any Attribute Node

Use the **Relaxation factor** field to specify a scalar relaxation factor that controls the damping of the block SOR smoothing steps. The allowed values of this factor are between 0 and 2 (default: 0.5).

Use the **Line based on** list to control if the lines of nodes are based on the **Mesh** (the default) or on a **Matrix**. If you select **Matrix**, also define the **Maximum line length** (default: 20). This values determines the maximum length of the lines in number of DOFs for each block.

Use the **Multivariable method** list to control the line updates:

- If **Uncoupled** is selected, each block SOR smoothing step updates a set of degrees of freedom with the same name that are located on a line.
- If **Coupled** is selected (the default), each block SOR smoothing step updates all degrees of freedom located on a line.

For smoothing of the turbulence variables K and ε, **Coupled** is recommended.

The **Blocked version** check box is selected by default and it uses a version of the SOR method that is optimized for parallel computations.

Settings With and Without a Coarse Solver

- If used with a Coarse Solver, select a **Termination technique**. These are described for the SCGS attribute under Settings When Used With Coarse Solver.
- If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2).

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations to perform after the SOR Line iterations have been performed (default: 1).

Use the **Relaxation factor** field to specify a scalar relaxation factor that controls the damping of the SSOR updates for the iterations specified in the **Number of secondary iterations** field. The allowed values of this factor are between 0 and 2 (default: 0.7).

SOR Vector

The **SOR Vector** node (SOR) handles settings for the SOR vector iterative method. This is a method of SOR type with added functionality useful for electromagnetics problems involving the $\nabla \times (a\nabla \times .)$ curl-curl operator and where you use *vector elements* (available primarily for electromagnetic wave simulations in the RF Module). In short, the added functionality consists of performing SOR iterations on an auxiliary linear system in addition to the ordinary SOR iterations. This node can be used as preconditioner/smoother together with the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver nodes. Also see The SOR Vector Algorithm.

MAIN

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR Vector algorithm to use. Select:

- **SSOR vector** to perform one ordinary SOR iteration on the main system followed by a number of SSOR iterations on an auxiliary (projected) system and then one ordinary SORU iteration. This is repeated in each SSOR vector iteration.
- **SOR vector** to perform one ordinary SOR iteration followed by a number of SOR iterations on an auxiliary system. This is repeated in each SOR vector iteration.
- **SORU vector** to perform a number of SORU iterations on an auxiliary system followed by one ordinary SORU iteration. This is repeated in each SORU vector iteration.

The algorithms perform these iterations to preserve symmetry as a preconditioner and also when used as symmetric presmoother and postsmoother in a multigrid setting.

Specify a scalar **Relaxation factor** ω. The allowed values of this factor are between 0 and 2 (default: 1).

The **Blocked version** check box is selected by default and it uses a version of the SOR Vector method that is optimized for parallel computations.

Use the **Variables** list to specify variables to be included in the auxiliary system of the SOR Vector method.

Settings With and Without a Coarse Solver

- If used with a Coarse Solver, select a **Termination technique**. These are described for the SCGS attribute under Settings When Used With Coarse Solver.
- If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2). Then use the **Number of secondary iterations** field to specify the number of iterations to perform the auxiliary system for each main iteration (default: 1).

State

The **State** node is an attribute node that handles settings for state variables. A state is composed of a set of ODE variables. Each state has a separate **State** node. This attribute is used together with the Dependent Variables node.

GENERAL

The **State Components** section display the variable names for the states components. Also, when internal extra variables are used, these are displayed here as **Internal Variables**.

Use the **Solve for this state** check box to control whether to solve for the state or not. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for its values are determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** operation node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.

A variable can still be solved for despite not being stored in output and vice versa.

SCALING

Control the scaling of a variable with the **Method** list.

Specifying a **Method** for a variable here overrides the **Method** selected in the **Scaling** section of the corresponding **Variables** operation node unless **From parent** is selected.

Select:

- **Automatic** to get an automatically determined scaling.
- **From parent** to use the scaling type selected in the **Method** list in the **Scaling** section of the corresponding **Variables** operation node.
- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling. Then enter a value in the **Scale** field.
- **None** to get no scaling.

Stationary Acceleration

The **Stationary Acceleration** subnode ( ) can be useful to accelerate the solution process for nonlinear problems with a time-periodic stationary solution. You can add it as a subnode to all Time-Dependent Solver nodes. Instead of time-marching the problem from start to finish, the Stationary Acceleration node solves for a number of periods and then extrapolates the solution forward in time based on the average solution and the average time derivative. This solution process is repeated until the average time derivative has reached steady state.

> The **Stationary Acceleration** node can be used to speed up the solution process for some types of plasma models but it is not used by default for any physics.

STATIONARY ACCELERATION

- In the **Variables** list, add the dependent variable for which you want to use stationary acceleration. Click the **Add** button (+) to open an **Add** dialog box that contains all dependent variables in the study. Select the variables that you want to add and then click **OK**. You can also delete variables from the list using the **Delete** button ( ) and move them using the **Move Up** (↑) and **Move Down** (↓) buttons.
- From the **Components** list, select the dependent variables for which the stationary acceleration performs the averaging and extrapolation. Select **All** (the default) to perform averaging and extrapolation for all variables, or select **Manual** to select the variables that you want to apply stationary acceleration from the list that appears.
- In the **Frequency** field, enter the frequency of the periodic solution. The default value, 13.56 MHz, is a frequency that is commonly used for plasma processes.
- In the **Stationary tolerance** field, enter the tolerance used to terminate the outer acceleration iterations, when the average time derivatives are small enough (default value: 0.01).
- In the **Number of extrapolation cycles** field, enter the number of periodic cycles used to extrapolate the solution (default value: 50). The higher this number is the more the solution process is accelerated, but at the same time the process can lead to an unstable acceleration iteration process.
- In the **Number of period averaging cycles** field, enter the number of cycles over which the stationary acceleration takes the average (default value: 5).
- In the **Number of smoothing cycles** field you enter the number of cycles that the stationary acceleration solves for in each iteration of the acceleration scheme (default value: 10). The average is taken over the last cycles.

Stop Condition

The **Stop Condition** node (STOP) stops the solver when any of the specified conditions are fulfilled. It is an optional attribute subnode to the Parametric and Time-Dependent Solver nodes.

> StopCondition in the *COMSOL API Reference Manual*

STOP EXPRESSIONS

Use the table to specify expressions for the conditions that define when the solver should stop. The solver evaluates the active expressions after each time step or parameter step. The setting in the **Stop if** column of each expression determines how it is evaluated. For **True (>=1)** (the default) the stepping stops if the real part is greater or equal to one, which is useful when entering logic expressions that evaluate to a Boolean true or false (`comp1.EndTerminal(comp1.phis)<2.4`, for example). For **Negative (<0)** the stepping stops if the real part of the expression becomes negative (`comp1.z_pnt-0.0004`, for example). Another example of a stop condition is `timestep<0.04`, which makes the solver stop when the internal time step drops below 0.04 s (when the time-dependent solver hits a sharp transient, for example). To use such logical expressions, use the **True (>=1)** setting.

STOP EVENTS

This section is available when the **Stop Condition** node is an attribute of the Time-Dependent Solver operation node. This is because events are only supported for time-dependent solvers.

Use the **Implicit event** table to specify at which events the solver should stop. All implicit events defined in the model automatically appear in the table. The stepping stops when any event marked as active is triggered.

OUTPUT AT STOP

Select an option from the **Add solution** list to make the solver additionally store the corresponding solutions before and after the stop condition was fulfilled. Select:

- **No** (the default) to not store any additional solutions. The last solution stored is the one normally stored by the solver before the stop condition was fulfilled.
- **Step before stop** to store the last step taken by the solver before the stop condition was fulfilled. No solutions are stored after this point even if they normally would be.
- **Step after stop** to store the solver step at which the stop condition was fulfilled. Any solutions up to this point are also stored as they normally would be.
- **Steps before and after stop** to store all solutions that would be stored by either of **Step before stop** and **Step after stop**.

Which solutions are normally stored by the solver depends on the **Times to store** setting for the Time-Dependent Solver node and on the **Parameters to store** setting for the Parametric node.

An example of using **Step before stop** would be to make sure to capture the last state of a simulation before a certain condition has been fulfilled, without having to store all of the solver steps up until this point. The setting **Step after stop** would similarly be used to capture the first state fulfilling a certain condition. When both the state before and after the condition are of interest, use the setting **Step before and after stop** to capture the transition. If the stop condition was fulfilled by the reinitialization effect of an implicit event, **Step before stop** stores the solution before reinitialization and **Step after stop** stores the solution after reinitialization.

Select the **Add warning** check box to specify that the solver adds a warning when the solver has stopped due to a stop condition.

Time Parametric

The **Time Parametric** node () is an attribute node that handles settings for parameter stepping to add parametric sweeps. For each set of parameter values, a time-dependent problem is solved.

This attribute can be used together with a Time-Dependent Solver or another time-dependent solver. The functionality is then similar to when Parametric is added as a subnode to a Stationary Solver, but continuation is not supported. The initial data is the same for all parameters.

TimeParametric in the *COMSOL API Reference Manual*

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to modify the parameter table and sweep type.

Use the **Sweep type** list to specify the type of sweep to perform. The **Specified combinations** type (the default) solves for a number of given combination of values, while the **All combinations** type solves for all combination of values. Using all combinations can lead to a very large number of solutions.

Use the **Parameter names** and **Parameter value list** table to specify parameter names and values for the parametric solver. Use the **Add** button (+) to add a row to the table. Each row has one parameter name and a corresponding parameter value list. For the **Specified combinations** sweep type, the list of values must have equal length. When you click in the **Parameter value list** column to define the parameter values you can click the **Range** button ( ) to define a range of parameter values.

If more than one parameter name has been specified the lists of parameter values are interpreted as follows: Assume that the parameter names are `p1` and `p2`, and that `p1` has the list 1 3 and `p2` has the list 2 4. For the **Specified combinations** sweep type, the solver first uses `p1` equal to `1` and `p2` equal to `2`. Thereafter, it uses `p1` equal to `3` and `p2` equal to `4`. And when the sweep type in **All combinations**, the solver uses the following order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. You can use the **Load from File** button (📂) to browse to such a text file. The read names and values are appended to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values, and a space separating the values. Click the **Save to File** button (💾) to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ *for* Excel®).

CLUSTER SETTINGS

Select the **Distribute parameters** check box to distribute the parameters on several computational nodes. If the problem is too large to run on a single node you can enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a non-distributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

Vanka

The **Vanka** node ( ) handles settings for the Vanka iterative method. Formally, this method applies to saddle-point problems (that is, problems where the equilibrium solution is neither a maximum nor a minimum) as a preconditioner/smoother. The corresponding linear system matrix is indefinite and its diagonal often contains zeros. A typical example is the Navier-Stokes equations. Problems formulated with weak constraints are also of this type. In short, the method can be described as a block SOR method. Local coupling of certain degrees of freedom (typically the Lagrange multiplier degrees of freedom) determines the blocks. Ordinary SSOR iterations are performed for degrees of freedom not involved in the block method. This attribute node can be used together with the Iterative, Krylov Preconditioner, Presmoother, Postsmoother, and Coarse Solver attribute node.

For more detailed information about the Vanka method, see The Vanka Algorithm.

MAIN

Settings With and Without a Coarse Solver

- If used with a Coarse Solver, select a **Termination technique**. These are described for the SCGS attribute under Settings When Used With Coarse Solver.
- If a Coarse Solver is not used, enter the **Number of iterations** to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2).

Settings When Used With Any Attribute Node

Use the **Variables** list to specify variables to include in a Vanka block approach.

Use the **Block solver** list to specify how to solve the Vanka block linear systems:

- Select **Direct** (the default) to use a direct solver.
- Select **Direct, stored factorization** to store the factorization. Storing the factorization makes the solver faster because the factorization is then not performed every update, but the storage uses more memory. If two Vanka smoothers are used as presmoother and postsmoother of a Multigrid solver, with similar enough settings, they share the same stored factorization, which means that they only use half the memory.
- Select **GMRES** to use the iterative method GMRES.

If you use the Vanka algorithm as preconditioner, or as smoother to a multigrid preconditioner when either of GMRES, Conjugate gradients, or BiCGStab is used as the linear system solver, use the **Direct** or **Direct, stored factorization** option in the **Block solver** list in order to get a stationary preconditioner.

The **GMRES** option can be useful if you use the FGMRES method as linear system solver because it can handle preconditioners that are not stationary. The **GMRES** option can also be useful if you use the Vanka algorithm as smoother to a multigrid solver because GMRES can be a bit faster than the direct solver.

When **GMRES** has been selected in the **Block solver** list the following options become available. Use the **Tolerance** field to specify the termination tolerance of GMRES (default: 0.02). Use the **Number of iterations before restart** field to specify how many iterations the solver should take between each restart (default: 100).

Use the **Relaxation factor** field to specify a scalar relaxation factor ω. The allowed values of this factor are between 0 and 2 (default: 0.8).

Select the **Blocked version** check box (selected by default) to use a version of the Vanka method that is optimized for parallel computations.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations to perform for degrees of freedom not involved in the Vanka blocks.

Use the **Relaxation factor** field to specify a scalar relaxation factor for the iterations specified in the **Number of secondary iterations** field (default: 1). The allowed values of this factor are between 0 and 2 (default: 1).

References for the Linear System Solvers and the Preconditions

1. http://graal.ens-lyon.fr/MUMPS/

2. www.netlib.org/linalg/spooles/

3. www.pardiso-project.org/

4. http://www.netlib.org/lapack/index.html

5. http://www.netlib.org/scalapack/index.html

6. https://computation.llnl.gov/casc/hypre/software.html

7. https://computation.llnl.gov/casc/hypre/download/hypre-2.9.0b_usr_manual.pdf

8. Greenbaum, A., "Iterative Methods for Linear Systems," *Frontiers in Applied Mathematics*, vol. 17, SIAM, 1997.

9. Y. Saad and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856–869, 1986.

10. Y. Saad, *Iterative Methods for Sparse Linear Systems*, Boston, 1996.

11. Y. Saad, "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM J. Sci. Statist. Comput.*, vol. 14, pp. 461–469, 1993.

12. M.R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Standards*, vol. 49, pp. 409–435, 1952.

13. C. Lanczos, "Solutions of Linear Equations by Minimized Iterations," *J. Res. Nat. Bur. Standards*, vol. 49, pp. 33–53, 1952.

14. H.A. Van Der Vorst, "A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. Statist. Comput.*, vol. 13, pp. 631–644, 1992.

15. J.R. Gilbert and S. Toledo, "An Assessment of Incomplete-LU Preconditioners for Nonsymmetric Linear Systems," *Informatic*a, vol. 24, pp. 409–425, 2000.

16. Y. Saad, *ILUT: A Dual Threshold Incomplete LU Factorization*, Report umsi-92-38, Computer Science Department, University of Minnesota, available from http://www-users.cs.umn.edu/~saad.

17. W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

18. R. Beck and R. Hiptmair, "Multilevel Solution of the Time-harmonic Maxwell's Equations Based on Edge Elements," *Int. J. Num. Meth. Engr.*, vol. 45, pp. 901–920, 1999.

19. S. Vanka, "Block-implicit Multigrid Calculation of Two-dimensional Recirculating Flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 59, no. 1, pp. 29–48, 1986.

20. H.C. Elman and others, "A Multigrid Method Enhanced by Krylov Subspace Iteration for Discrete Helmholtz Equations," *SIAM J. Sci. Comp.*, vol. 23, pp. 1291–1315, 2001.

21. A. Toselli and O. Widlund, "Domain Decomposition Methods—Algorithms and Theory," *Springer Series in Computational Mathematics*, vol. 34, 2005.

The COMSOL Multiphysics implementations of the algebraic multigrid solver and preconditioner are based on the following references:

22. K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD Report 70, GMD, 1999.

23. C. Wagner, *Introduction to Algebraic Multigrid*, course notes, University of Heidelberg, 1999.

24. R. Hiptmair, "Multigrid Method for Maxwell's Equations," *SIAM J. Numer. Anal.*, vol. 36, pp. 204–225, 1999.

25. D. J. Mavriplis, “Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flows,” *ICASE Report No. 98-7 (NASA/CR-1998-206911)*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1998.

26. V. John and G. Matthies, “Higher-order Finite Element Discretization in a Benchmark Problem for Incompressible Flows,” *Int. J. Numer. Meth. Fluids*, vol. 37, pp. 885–903, 2001.

27. V. John, “Higher-order Finite Element Methods and Multigrid Solvers in a Benchmark Problem for the 3D Navier-Stokes Equations,” *Int. J. Numer. Meth. Fluids*, vol. 40, pp. 775–798, 2002.

Solution Utility Nodes

The following sections describe the solver utility nodes (listed in Table 19-8) and the settings in detail.

TABLE 19-8: SOLUTION UTILITY NODES

ICON	NAME	DESCRIPTION
	Adaptive Mesh Refinement (Utility Node)	Stores the solution on the finest mesh from a mesh refinement procedure.
	Assemble	Provides a way of accessing assembled matrices and vectors for further work in Java®.
	Compile Equations	Compiles equations by specifying which study and study step to use and specifying representation of complex variables.
	For and End For	Add a for loop to the solver sequence to iterate some solver commands.
	Input Matrix	Used to create the raw data of an assembled matrix or vector from Java®.
	Store Solution	Stores the solution at this point of the solver configuration also after the solver configuration has been computed.
	State Space	Provides a way of accessing state-space matrices for further work in Java.

About Solver Commands in the *COMSOL API Reference Manual*

Adaptive Mesh Refinement (Utility Node)

The **Adaptive Mesh Refinement** ( ) utility node is added automatically (to an otherwise empty solver configuration) by the corresponding Adaptive Mesh Refinement attribute node. It is a container for a solution obtained using the attribute node. It is not possible to add this node manually and it does not have any settings.

Assemble

The **Assemble** node ( ) provides access to the raw data of any assembled matrix or vector. Right-click the solver node and select **Other>Assemble**. Select the appropriate check boxes for the matrices and vectors you want to inspect or modify and save the model as a file for Java. You can also display the values of these system matrices in a table using the System Matrix node ( ) found under **Results>Derived Values**. The saved Java file now contains code for assembling the selected matrices and vectors that can be used to access the matrix rows, columns, values, and so forth. For information about the eliminated system, see Elimination Constraint Handling.

Note that some study types require that additional parameters are defined. In order to get the expected matrices you have to specify these parameters manually. If you get an error about undefined variables you can define the variables in the Parameters node found under **Global Definitions**. Note that the value you set is the one that is used in the expressions where the variable is found. Some examples of variables that might be needed are

- `t`, the requested output time

- `timestep`, the time step used by the solver, for time-dependent problems
- `freq`, the frequency to assemble the problem for, for frequency-dependent problems.

There are also other variables that might be needed: `phase` (the phase), `niterCMP` (the nonlinear iteration number), and `CFLCMP` (a pseudo-time-stepping control variable).

ELIMINATED OUTPUT

In this section you can choose to output matrices and vectors that are passed to the linear solvers—that is, where constraints have been eliminated—by enabling one or several of the following check boxes: **Eliminated load vector, Eliminated stiffness matrix**, **Eliminated damping matrix**, **Eliminated mass matrix**, **Constraint null-space basis**, **Constraint force null-space basis**, **Particular solution (ud)**, and **Scale vector**.

NON-ELIMINATED OUTPUT

In this section you can choose to output matrices and vectors that the solver assembles before the elimination step by enabling one or several of the following check boxes: **Load vector**, **Stiffness matrix**, **Damping matrix**, **Mass matrix**, **Constraint vector**, **Constraint Jacobian**, and **Constraint force Jacobian**.

OPTIMIZATION OUTPUT

In this section you can choose to output matrices and vectors assembled during optimization by enabling one or several of the following check boxes: **Optimization constraint Jacobian**, **Optimization constraint vector**, **Lower bound constraint vector**, and **Upper bound constraint vector**.

ADVANCED

If you want to assemble an eigenvalue problem you can set the **Eigenvalue name** (default: `lambda`) and the **Value of eigenvalue linearization point** by first selecting the **Set eigenvalue name** check box.

LOG

This section, which is initially empty, contains a log from the assembling.

Assemble in the *COMSOL API Reference Manual*

Compile Equations

In the **Compile Equations** () node's settings window you specify which study and study step to use when computing the current solver configuration and compiling the equations to solve. The node displays the name of the selected study step: **Compile Equations: Stationary**, for example. Right-click this node and select **Statistics** to see the number of degrees of freedom for the solver (see The Statistics Page).

STUDY AND STEP

Specify the study in the **Use study** list and the study step in the **Use study step** list. By default you get the parent study and its first study step.

Complex variables are by default represented by complex-valued degrees of freedom. By selecting the **Split complex variables in real and imaginary parts** check box, the representation of complex variables is changed to using separate real degrees of freedom for the real and imaginary parts. The split representation can improve convergence where nonanalytic functions of complex variables are used in equations. Using a split representation also makes it possible to avoid complex pollution (a small nonzero imaginary component) of real variables by specifying a *real* or *complex* value type for variables. If you use a split representation of complex variables, specify the value type of dependent variables in the **Discretization** sections in the settings windows for the main physics nodes.

The split representation enables a correct evaluation of Jacobians for the following operators: `real`, `imag`, `conj`, `abs`, and `realdot`.

For and End For

From the **Programming** submenu for solver sequences, you can select **For** to add two nodes last to the sequence, a **For** node ( ) and an **End For** node ( ). You can then move these node where you want to create a for loop that iterates some part of the solver sequence. When the sequence is run, the for loop runs the node in between the **For** and **End For** nodes a fixed number of steps. You can add more than one **For** node to create nested for loops. The model tree displays the loop structure by indenting the description text for the nodes. The for loops must be balanced; otherwise and error occurs. For loops can be useful, for example, for solving particle-field interactions in particle tracing by iterating between a stationary and a time-dependent solver.

The **End For** node has no settings. The settings window for the **For** node contains the following setting:

GENERAL

Use the **Number of iterations** text field to specify for how many times to run the solver nodes between the **For** node and the enclosing **End For** node (default value: 5). Any nonnegative integer is supported (including zero).

The Statistics Page

Use the **Statistics** page to view statistics about a solver, its dependent variables, and their number of degrees of freedom. This can be done before solving the problem and is useful for determining which variables are the most costly to compute and store. To open the **Statistics** page for a solver, right-click the Compile Equations node ( ) and select **Statistics**. This page contains the following section:

NUMBER OF DEGREES OF FREEDOM

Here you see a list of the dependent variables and their number of degrees of freedom (DOFs) as well as the total number of DOFs. The list includes both variables solved for and variables not solved for.

Computing this statistic requires a computation of the size of the assembled finite element model. This can take some time for large models.

Input Matrix

The **Input Matrix** node ( ) can be used to create the raw data of an assembled matrix or vector from Java. Right-click the Eigenvalue Solver, Stationary Solver, or Time-Dependent Solver nodes and select **Input Matrix**. Select the appropriate check boxes for the matrices and vectors you want to input and save the model as a file for Java. The saved Java file now contains code for inputting the selected matrices and vectors.

INPUT

In this section you can choose to input matrices and vectors that are passed to the linear solvers by enabling one or several of the following check boxes: **Load vector, Stiffness matrix**, **Damping matrix**, **Mass matrix**, **Constraint vector**, **Constraint Jacobian**, and **Constraint force Jacobian**.

The matrices and vectors input using the **Input Matrix** node replace the corresponding matrices and vectors in the assembled system from the model in COMSOL Multiphysics.

- Elimination Constraint Handling
- InputMatrix in the *COMSOL API Reference Manual*

Store Solution

This is a utility node whose purpose is to make it possible to access intermediate solution results. By default, the software only stores the solution at the last computed node of a solver configuration. Add a **Store Solution** node ( ) anywhere in a solver configuration to make the software store the solution at that point of the sequence in addition to the solution at the last computed node. You can use the solution from a **Store Solution** node when analyzing the results and as initial values (via a study-type node) for other computations.

GENERAL

This section contains the name of the solver sequence that the **Store Solution** node is using.

LOG

This section is initially empty. It contains information if the **Store Solution** node is used to store parametric sweep data.

StoreSolution in the *COMSOL API Reference Manual*

State Space

The **State Space** node ( ) provides access to the raw data of a PDE in state-space form. To create state-space data, right-click the solver node and select **Other>State Space**. Specify the input and output and the state-space matrices and vectors that you want to access. Then save the model as a file for Java. The saved Java® file now contains code for assembling the selected matrices and vectors that can be used to access the matrix rows, columns, values, and so forth.

INPUT

In the **Input parameters** field, enter all parameters that affect the model as space- or comma-separated entries.

OUTPUT

The state-space node assembles matrices that describe a model as a dynamic system when **Off** is selected from the **Static** list:

$$\begin{aligned} Mc\dot{x} &= MAx + MBu \\ y &= Cx + Du \end{aligned}$$

If you select **On** from the **Static** list, a static linearized model of the system is described by

$$y = (D–C(MA)^{-1}MB)u$$

In the **Output expressions** field, enter all expressions that are to be evaluated as output from the model as space- or comma-separated entries. Select any of the **MA**, **MB**, **D**, and **C** check boxes, and if **Static** is set to **Off**, any of the **MC**, **Null**, **ud**, or **x0** check boxes. **Null** is the PDE constraint null-space matrix, and **ud** is a particular solution fulfilling the constraints. **x0** is the initial data. The solution vector U for the PDE problem can then be written

$$U = \mathrm{Null}x + ud + u0$$

where $u0$ is the linearization point, which is the solution stored in the sequence once the state-space export node is run. The input linearization point is stored in the sequence after the state-space node is run.

LOG

This section, which is initially empty, contains a log from the run of the **State Space** node. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is available also when reopening the model.

StateSpace in the *COMSOL API Reference Manual*

Job Configurations

The **Job Configurations** node ( ) automatically displays if the Job node has content. Or to make it available in the context menu, click the **Show** button ( ) and select **Advanced Study Options**.

There are three categories of job configurations, each of which correspond to a node:

- Parametric
- Batch
- Cluster Computing

Parametric and Batch share the common set of subnodes listed in Table 19-9. In addition, a Batch job configuration node has a default Batch Data subnode that in turn stores External Process nodes, which contain information about batch jobs. Cluster Computing job configuration nodes do not have any subnodes and must point to a Batch job configuration.

- Using a Job Configuration to Store Parametric Results on File
- Advanced Study Extension Steps

TABLE 19-9: PARAMETRIC AND BATCH SUBNODES

ICON	NAME	DESCRIPTION
8.85 e-12	Derived Value	Runs a Derived Values node.
	Export to File	Runs an Export node and saves it to a file.
	External Class	Runs the main method of an external compiled Java® class.
	External Process	A Batch subnode that contains information about the batch processes that have been started by the Batch (Job Configuration) node. Each External Process node is associated with a started batch job.
	Geometry Sequence	Runs a geometry sequence.
	Job	Runs a Parametric, Batch, or Cluster Computing job node.
	Meshing Sequence	Runs a mesh sequence.
	Plot Group	Runs a Plot Group node.
	Save Model to File	Stores a model in the state that it is at that point in the job configuration.
	Solver	Runs a solver configuration.

TABLE 19-10: JOB CONFIGURATIONS NODE - CONTEXT MENU OPTIONS

ICON	NAME	DESCRIPTION
	Show Default Solver	Shows the default job configuration node (if any) that corresponds to the study step nodes in the study.
	Parametric (Job Configuration)	Adds a Parametric node, which can loop over a given set of parameters. For each set of parameters, it runs the sequence defined by its subnodes. You can combine the sequence with other Batch, Parametric, or Cluster Computing sequences in a hierarchical way by adding a job configuration that points to another node.
	Batch (Job Configuration)	Adds a Batch node to run batch jobs. The Batch job has a special Batch Data subnode that collects External Process subnodes containing job status information.
	Cluster Computing (Job Configuration)	Adds a Cluster Computing node, which is useful when you want to submit a batch job to a job scheduler want to run the model in distributed mode as a batch job.
	Delete Solvers	Deletes all jobs under the Job Configurations node.

* If you have the Optimization Module, Optimization is also an option.

Parametric (Job Configuration)

One of the main nodes is the **Parametric** ( ) job configuration, which can loop over a given set of parameters. For each set of parameters it runs the sequence defined by its subnodes. You can combine the sequence with other Batch, Parametric, or Cluster Computing sequences in a hierarchical way by adding a job configuration that points to another node. You can, for instance, create a **Parametric** node that does a LiveLink™ update and then runs a **Cluster Computing** node that in turn runs a second **Parametric** sequence on another node.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

You define the parameters in the **Parameter names** and **Parameter values** fields. The parameters can be loaded from file by clicking the **Read File** button when you have selected the file through the **Load Parameter Values** dialog box, which you open by clicking **Browse**. You can add a **Stop condition** that is evaluated after each solution. Once the condition evaluates to a negative value the **Parametric** node is stopped.

RESULTS WHILE SOLVING

Check the **Plot** check box to allow plotting of results while solving. Select what to plot and when from the **Plot group** list. The data set of the selected plot group is plotted as soon as the results become available.

Use the **Probes** list to select any probes to evaluate. Use the **Accumulated probe table** to accumulate data during a sweep. The accumulation is over solver variation (time, frequency, and so forth) and variation over the parametric sweep. For independent variation of parameters you can use the accumulated table with the **Format: Filled** to change the table data into a matrix format that can be used for response surface plots.

ERROR

Errors are usually stored in the **Error** table. If you want to get the error message at once, select the **Stop if error** check box.

CLUSTER SETTINGS

You can distribute the sweep on several computational nodes by selecting the **Distribute parameters** check box. If the problem is too large to run on a single node, enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a non-distributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

An Integro-Partial Differential Equation: model library path **COMSOL_Multiphysics/Equation-Based_Models/integro_partial**.

Using a Job Configuration to Store Parametric Results on File

Batch (Job Configuration)

The **Batch** ( ) job configuration is the main node for running batch jobs. Batch jobs run in separate processes. You can therefore continue working in the COMSOL Desktop once a batch job is run. Model changes in the COMSOL Desktop after the batch job is submitted do not affect the model in the batch job. The **Batch** job has a special Batch Data subnode, which collects External Process subnodes containing job status information.

As in the Parametric (Job Configuration) node the batch job is defined by a number of subnodes. The batch job then runs each subnode. Use the **Save as Default** button in the toolbar to save the current directory setting as the default directory for batch files.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Set the number of cores that the batch job should use in the **Number of cores** field. The default behavior is to use all available cores. If you set **Number of simultaneous** jobs to more than 1, several jobs can run at once. When you are running more than 1 job at once it is important that the product of **Number of cores** and **Number of simultaneous** jobs does not exceed the number of cores available on the computer. Otherwise you experience performance degradation. When you run multiple batch jobs on your computer COMSOL Multiphysics makes sure this does not happen if you are using the automatic setting. You can set a **Start time** if you want the batch process to start a later time. Select the hour to start the run in **Start time**. Select the **Use graphics** check box when the batch process should run results nodes that creates graphical contents such as exporting to file. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before is must inform that it is still running. Failure to do so means that the process is considered dead and a new process is started if the maximum number of job restarts is not reached.

FILES

Set the **Filename** of the model. If the batch job is generated from a parametric sweep, a unique name that depends on the parameter names and values is created. The default is to overwrite any previous models with the same name.

Disable the default by clearing the **Clear previous model** check box. Select the **Clear meshes** check box to clear the meshes before running the batch sweep. The default is to not clear the meshes. Select the **Clear solutions** check box to clear the solutions before running the batch sweep.

Specify the **Directory** to store the model. Click **Browse** if you want to browse to a directory.

- If you are connected to a COMSOL server on another computer, you can control the working directory used by the COMSOL server if you select the **Specify COMSOL server directory path** check box and enter the path to the server **Directory** or **Browse** for the path. Otherwise a temporary directory on the COMSOL server is used to save files.
- If the batch job has another path to the directory you select the **Specify external COMSOL batch directory path** check box and enter the path to the batch **Directory** or **Browse** for the path. If the batch job has another path to the directory you select the **Specify external COMSOL batch directory path** check box and enter the path to the batch **Directory** or **Browse** for the path.
- If COMSOL is installed in a different directory from where the batch job runs, enable the **Specify external COMSOL installation directory path** and specify the install directory (click **Browse** or enter the path to the **Directory**). This can occur if you are submitting jobs to a job scheduler with the **Cluster Computing** node.

Click the **Save As Default** button (💾) at the top of the settings window to save the current directory setting as the default directory for batch files.

SYNCHRONIZATION

Select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional postprocessing after the sweep has finished. The default is to disable solution synchronization. Select the **Synchronize accumulated probe table** check box to synchronize the accumulated probes computed by the batch processes with the model. The accumulated probe synchronization is enabled by default. Select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases, use the solution synchronization and probe synchronization functionality instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently. Use the **Probes** list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☒), and **Add** (+) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.

CLIENT SETTINGS

The **Batch** node can also be used as a client to drive a server on another machine. You enable the client functionality by selecting the **Client** check box. You can then set the **Host name** (default: localhost) and **Port number** of the server for the batch job (default: 2036) to connect to. This number is the default port number. If the server you want to connect to is using another port, then edit this number accordingly.

Micromixer—Batch Version: model library path **COMSOL_Multiphysics/Tutorial_Models/micromixer_batch**.

The COMSOL Commands

Cluster Computing (Job Configuration)

The **Cluster Computing** ( ) job configuration is useful when you want to submit a batch job to a job scheduler want to run the model in distributed mode as a batch job. When you have specified the cluster computing settings, click the **Save as Default** button ( ) in the settings window's toolbar to save the current setting as default.

These settings are saved to The Preferences Dialog Box in the **Multicore and Cluster Computing** section.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

If you want to use a job scheduler or do some other operations before or after the job you set the command line in the **Prepend command** and **Postpend command** fields. You can define a default command line with the system properties `cs.precmd` and `cs.postcmd`. If the command line contains `{nn}` or `{perhost}` they are replaced by the values in the **Number of nodes** field and **Number of processes** field, respectively.

From the **Batch job** list, select which batch job to submit. Click the **Go to Source** button) to move to the settings window for the selected **Batch** node.

CLUSTER SETTINGS

After making these settings, click the **Save as Default** ( ) button on the settings window toolbar to save the current directory settings as the default preference.

Choose the **Cluster type**—**General** (the default), **HPCS 2008**, **WCCS 2003**, **OGS/GE**, or **Not distributed**:

General

Select **General** (the default) to configure to run on many types of clusters and schedulers, including Linux clusters.

- When **General** is selected, and you have started a multiprocessor daemon (MPD) on the computer, click to select the **MPD is running** check box.
- The entry in the **Host file** field specifies the host file used for the job. If left empty, MPD looks for a file `mpd.hosts` in the Linux home directory.
- Select which bootstrap server should be used by MPI using the **Bootstrap server** setting.
- If your cluster is Linux and it requires that an SSH (secure shell) or RSH (remote shell) is installed in an uncommon directory, use the **Rsh** field to set the RSH communication protocol.
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter the **Number of processes on host**. The default is 1.

HPCS 2008

Select **HPCS 2008** to use the Windows HPC Server 2008 job scheduler to submit the batch job.

- If you want to include scheduler arguments, add them to the **Additional scheduler arguments** field (for example, for `mpiexec`).
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (the default is 1 node).
- Select a **Node granularity**—**Node** (the default), **Socket**, or **Core**. **Node** allocates one process on each host, **Socket** allocates one process on each socket, and **Core** allocates one process on each core.
- The **Exclusive nodes** check box is selected by default. Click to clear if you want to run on nodes shared by other users.

Under **Advanced**:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- Set the names of **Requested nodes**. The job scheduler only allocates jobs on the nodes listed by you.
- Enter the **Node group**. The job scheduler only allocates jobs on the nodes belonging to the group.
- Enter the minimum required **Cores per node**. The default is 0. The job scheduler only allocates jobs to nodes with at least as many cores as set.
- Enter the minimum required **Memory per node (MB)**. The default is 0. The job scheduler only allocates jobs to nodes with at least as much memory as set.
- Enter the **Runtime (minutes)** before the job is canceled. The default is **Infinite**.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Select a **Priority**—**Highest**, **Above normal**, **Normal** (the default), **Below normal**, or **Lowest**—for the scheduled job.

WCCS 2003

Select **WCCS 2003** to use the Windows Compute Cluster Server 2003 job scheduler to submit the batch job.

- If you want to include scheduler arguments, add them to the **Additional scheduler arguments** field (for example, for `mpiexec`).
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (the default is 1 node).
- The **Exclusive nodes** check box is selected by default. Click to clear as required.

Under **Advanced**:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- Set the names of **Requested nodes**.
- Enter the **Runtime (minutes)** before the job is canceled. The default is **Infinite**.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Select a **Priority**—**Highest**, **Above normal**, **Normal** (the default), **Below normal**, or **Lowest**—for the scheduled job.

OGS/GE

Select **OGS/GE** to use the Open Grid Scheduler/Grid Engine job scheduler to submit the batch job.

When **OGS/GE** is selected:

- If you want to include scheduler arguments, add them to the **Additional scheduler arguments** field (for example, for `mpiexec`).
- Select the **Bootstrap server** that should be used by MPI.
- If your cluster is Linux and it requires that an SSH (secure shell) or an RSH (remote shell) is installed in an uncommon directory, use the **Rsh** field to set the RSH communication protocol.
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Select a **Slot granularity**—**Host**, **Slot**, or **Manual**—to specify if COMSOL should parallelize on the physical **Host** level or on the OGS/GE-allocated **Slot** level. For **Host** and **Slot** specify the **Number of slots** to allocate**.** The **Manual** setting can be used to control the granularity more. In this case set the number of computational nodes to use in the **Number of nodes**. For **Slot** and **Manual** the number of processes on each node is set in the **Number of processes on host** field; usually this is 1.
- Enter the **Queue name** to set the name of the Sun Grid Engine.
- The Sun Grid Engine priority is set in the **Priority value** field. The default is 0.

Not Distributed

Select **Not distributed** when you want to submit a batch job to a job scheduler without running a distributed job.

REMOTE AND CLOUD ACCESS

See Remote and Cloud Access described for Cluster Computing.

Micromixer—Cluster Version: model library path **COMSOL_Multiphysics/Tutorial_Models/micromixer_cluster**.

Using a Job Configuration to Store Parametric Results on File

As an example of the use of a job configuration, right-click the **Job Configurations** ( ) node and add a Parametric (Job Configuration) to run a parametric study, storing the model and associated data and plots for each parameter step to individual files. This is useful if you, for example, want to:

- Avoid large model files while running large parametric sweeps.
- Store information in individual output files instead of in one large file.
- Control several file outputs directly from the COMSOL Desktop: Model MPH-files, data files (text files), and image files.

The following steps describe the most important parts of setting up a job configuration to accomplish the desired file output:

1. Start with a model that does not contain any parametric sweep and define your outputs. Then add them to the **Export** node (by right-clicking and adding **Data**, **Plot**, and image nodes as desired) and assign each data or plot output to a file.
2. Create a parametric sweep study by right-clicking the study node and adding a **Parametric Sweep**. Add the parameters you want to sweep over in Parameters under Global Definitions. Now add the parameters in **Parameter names** in the parametric sweep node you created and set the **Parameter values** you want to sweep over. In some cases COMSOL Multiphysics chooses to use the more efficient parametric solver when sweeping. Set the **Use parametric solver** under **Study Extensions** to **Off** if you want to avoid this. Use the **Keep solutions in memory** setting **Only last** to conserve memory. In that case use the Accumulated Probe table to store the data you want to use for later processing. You can also choose to store the models on file by enabling the **Save each solution as**

model file and then enter a filename in the **Filename** field, or click **Browse** to choose a name and location for the model files.The models created during the simulation can then be found in a **Save model to File** node under **Job Configurations**. The default settings are configurable in the Preferences dialog box.

3 If you want to create output files with the parametric sweep for further analysis, right-click the main study node and select **Show Default Solver**. Select the parametric node under **Job Configurations** that corresponds to the parametric sweep you want to export data from. You can see which parametric sweep the node corresponds to from the **Defined by study step** list. Note that if you run a stationary study, you need to switch off **Use parametric solver** in the **Study Extensions** section (see above) to get the node under **Job Configurations**.

   - Right-click the **Parametric** node and select **Results>Plot Group** if you want to run a **Plot Group** for each parameter value in the sweep. In most cases use the **Plot** settings under **Results While Solving** (in the study step's settings window) instead.
   - Right-click the **Parametric** node and select **Results>Derived Value** to run a **Derived Values** node for each parameter value in the sweep. This functionality is similar to probes but is useful if you have already set up a **Derived Values** node that you want to use during the sweep. The values are stored in a table (similar to probes) for further processing.
   - Right-click the **Parametric** node and select **Results>Export to File** if you want to export data to a file for each parameter value in the sweep, running an **Export** node under **Results**. Note that after the sweep, the files created are listed in the node and can be opened in a browser using the **Open** button.

Advanced Job Configurations

Because a job configuration defines a sequence of steps you can create highly advanced models that use different solvers and sweeps as input to each other and performs different types of postprocessing during a sweep. Here are some suggestions:

- Create a parametric job configuration that uses two solvers. The first solver is used as input to the second solver. One way to create such a sweep is to create two studies. Let the second study use the first as input from the **Values of Dependent Variables**. Create a **Parametric Sweep** in the first study and select to keep **Only last** solution. Run **Show Default Solver** in both studies and enter the **Job Configurations** node of the first study. Right-click the **Parametric** node and add a **Solver**. Select the new solver and set it to run the second solver from the second study. Also check the **Keep all solutions** settings. Right-click the **Parametric** node and select **Run**. You can add further functionality to the sequence such as **Export to File** to suit your needs.
- Another possibility is to create a parametric sweep that runs a **Cluster Computing** node. This gives you similar functionality as the distributed parametric sweep, but the results are stored in separate files. Note that this requires several licenses, one for each process running simultaneously. One way to create this is to add a **Cluster Sweep** node by right-clicking the study. This node creates such a sweep automatically. It also sets up synchronization of solutions and accumulated probe tables when the synchronizations is enabled. Another, more complicated, way is to add a **Cluster Computing** node by right-clicking the study. Note that you need the **Advanced Study Options** enabled. If you added the **Cluster Computing** node then run **Show Default Solver** and select the **Job Configurations** node. Right-click and add a **Parametric** node. Right-click the new **Parametric** node and enable it. Right-click again and add a **Job** node. Set the job sequence to point to the **Cluster Computing** node. This sets up the new **Parametric** node to start a new process for each parameter. After you have set the parameters that you want to compute for, right-click and select **Run**. The resulting models are stored in **External Process** nodes under the batch job run by the cluster computing node. In order to get exported data during the runs you can use the functionality **Export to File** described above or enable the synchronization of solutions and accumulated probe tables in the **Batch** node.
- Create a parametric sweep that runs a solver and a class file that uses the COMSOL API to modify the solution. This can be useful if you want COMSOL to communicate with another program. You do this by adding a **Parametric Sweep** node to the study that you want to use. Run **Show Default Solver** and enter the **Job Configuration**s

node. Right-click the **Parametric** node and select **Other>External Class**. You can also modify the **Parametric** node by, for example, adding a **Stop condition**.

Batch Data

The **Batch Data** node ( ) contains information about the batch processes that have been started by the Batch (Job Configuration) node ( ). Each process is represented by an External Process ( ) subnode. On the Batch Data settings window, click the:

- **Attach Job** button ( ) on the settings window toolbar to display the progress of all the external processes. The GUI enters a progress mode in order to follow the progress of the external processes.
- **Stop all Processes** button ( ) to send the stop command to unfinished jobs.
- **Cancel all Processes** button ( ) to send the cancel command to unfinished jobs.

Micromixer—Batch Version: model library path **COMSOL_Multiphysics/Tutorial_Models/micromixer_batch**.

Derived Value

The **Derived Value** node ( ) runs a Derived Values node defined in the **Results** branch of the model tree. To add this node, right-click the Parametric (Job Configuration) or Batch (Job Configuration) node and select it from the **Results** submenu.

The computed value is stored in a **Result** table under the **Derived Value** node (see Derived Values and Tables).

GENERAL

You select the **Derived Values** node to run from the **Run** list. The default behavior (**All**) is to run all **Derived Values** nodes.

RESULT

The **Table** setting decides which **Table** under **Results** to store the computed values in. The default is **New** for a new table. Clear the **Clear previous** check box if you want to add the computed values in each step of the job instead of clearing the values from an earlier sweep. This is useful if you want to add the values for additional parameters to an already existing table. The **Parameters** column in the **Result** table contains the parameters that computed the numerical value, the **Value** column contains the numerical value, and the **Derived values** column contains the name (tag) of the **Derived Values** node that computed the numerical value. The information about the derived values nodes is useful when you have selected **All** from the **Run** list and the computed values come from different **Derived Values** nodes.

Export to File

The **Export to File** node ( ) runs an Export node defined in the **Export** branch of the model tree. The file is stored with a unique name that is generated from the current parameter values and the filename that the Export node has set. To add this node, right-click the Parametric or Batch job configuration node and select it from the **Results** submenu. Also see Exporting Data and Images.

GENERAL

Use the **Run** setting to select the Export node to run. The default behavior (**All**) is to run all Export nodes.

FILE

In the **File** section you set if COMSOL should overwrite files with the same name or if an error should occur, for example. If you clear the **Clear previous** check box, the job adds the values in each run instead of clearing the previous value. If you clear the **Add parameters to filename** check box, COMSOL does not create unique filenames for the current parameter values. This is useful if you want to start batch jobs with different parameter values from the command line and use the resulting file for further postprocessing.

OUTPUT

In the **Output** section you find the names of the files created during a sweep. You can select to **Open** a file by clicking the button. The file then opens in a web browser. The **Parameters** column in the **Output** table contains the parameter names, and the **Filename** column contains the corresponding filename.

External Class

The **External Class** node () runs the main method of an external compiled Java class file. To add this node, right-click the Parametric (Job Configuration) or Batch (Job Configuration) node and select it from the **Other** submenu.

GENERAL

Before the external class is called the system property `cs.currentmodel` is set to the name of the model that is calling the external class. You can set the name of the class file in the **Filename** field. Arguments can be passed to the main method with **Input**.

External Process

The **External Process** nodes () under a Batch Data node contain information about the batch processes that have been started by the Batch (Job Configuration) node. Each **External Process** node is associated with a started batch job.

On the External Process settings window, you can click these buttons, which are also available on The External Process Window where you can monitor this node’s progress.

- **Attach Job** button () on the settings window toolbar to display the progress of all the external processes. The GUI enters a progress mode in order to follow the progress of the external processes.
- **Stop all Processes** button () to send the stop command to unfinished jobs.
- **Cancel all Processes** button () to send the cancel command to unfinished jobs.
- **Clear Status** button () to clear the status of the selected job. Useful when the status indicates that the process is running but the process has failed.
- **Rerun Job** () to restart the selected job.

After clicking the button, the status of the requested operation is viewed in the Process Status section.

GENERAL

The **Start command** field contains the command that was used to start the batch job. The **Filename** contains the filename of the model that is used in the batch job. Click **Open** when the batch job has finished to open the file.

PARAMETERS

Parameter names and **Parameter values** are listed in this section when available.

PROCESS STATUS

The log is updated when you choose **Update log**.

Geometry Sequence

The **Geometry Sequence** node ( ) runs a geometry sequence. To add this node, right-click the Parametric (Job Configuration) or Batch (Job Configuration) node and select it from the **Other** submenu.

GENERAL

Select the sequence to **Run** from the list. The default is to run **All** geometry sequences. If a specific geometry sequence is chosen, you can click the **Go to Source** button ( ) to go to the **Geometry** node containing this sequence.

Job

You can add a **Job** node ( ) to run another job configuration. Right-click a Parametric (Job Configuration) or Batch (Job Configuration) to add the **Job** node.

GENERAL

Select the sequence to **Run** from the list. Recursive calls are detected and cause errors. Click the **Go to Source** button ( ) to go to the source job configuration node.

Meshing Sequence

The **Meshing Sequence** node ( ) runs a mesh sequence. To add this node, right-click the Parametric (Job Configuration) or Batch (Job Configuration) node and select it from the **Other** submenu.

GENERAL

Select the sequence to **Run** from the list. The default is to run **All** meshing sequences. Click the **Go to Source** button ( ) to go to the source mesh sequence.

Plot Group

The **Plot Group** node ( ) runs a sequence of plot groups, creating a plot in the Graphics window. To add this node, right-click the Parametric (Job Configuration) or Batch (Job Configuration) node and select it from the **Results** submenu. Also see Plot Groups and Plots.

GENERAL

Select the sequence to **Run** from the list. The default is to run **All** plot group nodes. Click the **Go to Source** button ( ) to go to the source plot. You can use an Export to File node to store the resulting plot in a file.

Save Model to File

The **Save Model to File** node ( ) stores a model in the state it is at that point in the Batch or Parametric job configuration.

GENERAL

The **Overwrite previous model files** check box is selected by default. This means that previous models with the same name are overwritten. If a parametric sweep is running, the model is given a unique name based on the current parameter values. The **Add parameters to filename** check box is selected by default. Click to clear it to use the same name for each parameter value. Enter a **Filename** including its network path, or click **Browse** to navigate to the location on your network where you want to store the model.

OUTPUT

The names of the saved models are stored in the table under **Output** where the **Filename** and **Parameters** are listed. Open a saved model in a new instance of COMSOL Multiphysics by selecting an **Open file** from the list or by clicking the **Open** button.

Solver

The **Solver** node ( ) runs a solver configuration for either a Parametric (Job Configuration) or Batch (Job Configuration).

GENERAL

Select the sequence to **Run** from the list. The default is to run **All** solver configurations. Click the **Go to Source** button ( ) to go to the source solver configuration.

OUTPUT

Store a copy of the solution once it has run by selecting the **Keep all solutions** check box. The name of the copy is generated from the current sequence name and parameter values. The default is to clear previous solutions. To disable it, clear the **Clear previous** check box. To add solver nodes to a solver configuration, right-click a **Solver** node and then select a solver from the **Solvers** submenu.

SOLUTION

This section contains a table of parameters and solutions after running a batch job or a parametric sweep. In the **Parameters** column you find the parameters in the parametric sweep, for example, and the values for those parameters at that step in the sweep. The corresponding row of the **Solution** column contains the name of the solution that corresponds to that set of parameter values. Typically, **Store Solution** nodes store those solutions, which are also available from the **Solution** list in the Solution data sets.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

In general, how the perturbation concept (and the study) is named is based on the application. For example, in the AC/DC Module and Semiconductor Module it is referred to as *small-signal analysis*, whereas in the Structural Mechanics Module it is referred to as *prestressed analysis*. For the Batteries & Fuel Cells Module and the Electrodeposition Module, the studies are called *AC impedance*. For the CFD Module, and fluid flow in general, the term *perturbation* is sufficient.

See these study types for details about availability by module and physics interface:

- Frequency-Domain, Perturbation
- Small-Signal Analysis, Frequency Domain
- Prestressed Frequency Analyses Studies
- AC Impedance Stationary
- AC Impedance Time Dependent

- If you have the Acoustics Module and AC/DC Module, see Loudspeaker Driver: model library path **Acoustics_Module/Industrial_Models/loudspeaker_driver**.
- If you have the AC/DC Module, see Small-Signal Analysis of an Inductor: model library path **ACDC_Module/Inductive_Devices_and_Coils/small_signal_analysis_of_inductor**.

Frequency-Domain, Perturbation Study and Study Step

Usually, two different right-hand-side contributions (or loads) must be defined for each step. The first step needs a stationary value for the contribution, and the second step needs the perturbed value for the contribution. The definition of these contributions differs between exclusive and contributing nodes, and this relates to the **Harmonic Perturbation** node, which can be added to a wide variety of physics interface nodes (for example, the **Electric Potential** and **Electric Ground** nodes for the **Electric Currents** interface).

For plot settings made available by using this study, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

If you have the Acoustics Module and AC/DC Module, see Axisymmetric Condenser Microphone with Electrical Lumping: model library path **Acoustics_Module/Industrial_Models/condenser_microphone_lumped**.

Harmonic Perturbation—Exclusive and Contributing Nodes

A physics node that is exclusive has a **Harmonic Perturbation** subnode. This subnode adds harmonic perturbations to the right-hand-side contributions of its parent node (for example, a **Boundary Load** on the **Solid Mechanics** interface or a **Terminal** node on the **Electric Currents** interface). In the settings window, the perturbation is entered for these contributions, which is only used when you solve for a **Frequency-Domain, Perturbation** study type. The parent node defines the stationary value for the contribution, which is not present for the **Frequency-Domain, Perturbation** study.

Harmonic perturbation nodes have a tilde over the top of the node, as in this example of a boundary level node ( ).

Contributing nodes have a yellow dot to the left of the node, as in this example of a boundary level node ( ).

Exclusive nodes are of two types—Override ( ) and Overridden ( )— and in both types a red arrow is used at the top (override) or bottom (overridden) left corners.

Nodes that are contributing (typically sources) can add their contributions as a harmonic perturbation. To define the stationary value for the contribution, you can add another node of the same type with the harmonic perturbation setting cleared.

For exclusive loads there is only one way of doing it due to the exclusivity—as a subnode. This subnode cannot, in general, be a full copy of the original node because only some subsets of data can be changed.

As an example (prescribed displacement in structural mechanics), the prescribed displacement must have the same local system and the harmonic perturbation can only be applied to degrees of freedom already prescribed in the parent node.

Also, all contributing nodes are free to use the full set of settings. A static point load can be at one point in the global direction, and a local system for the harmonic contribution can be used.

- Physics Exclusive and Contributing Node Types
- For different plot settings made available, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

Vibrating MEMS structures are often prestressed. For example, a cantilever structure could be prestressed by applying a DC voltage bias between the cantilever and a nearby ground plane, then vibrations could be driven at resonance by applying an additional AC bias. Another common example would be a clamped-clamped beam with a residual thermal stress.

20

Results Analysis and Plots

This chapter describes the functionality for visualization and analysis of simulation results in COMSOL Multiphysics®.

Postprocessing and Analyzing Results

The **Results** branch in the COMSOL Multiphysics model tree contains tools for postprocessing and analyzing the results from your simulations, including visualizations, animations, and data analysis. The **Results** branch groups the tools into the following categories:

- Data Sets contain the source of data for plotting, for example, by indicating a solution and geometry or by transforming another data set (for combining solutions or evaluating data along a cut line, for example).
- Plot Groups and Plots: A plot group is a collection of plots to display simultaneously in the **Graphics** window. The plot groups include 1D plots (graphs), 2D plots (surface plots, for example), and 3D plots (volume plots, for example) with many different plot types and options. You can enable or disable plots in a plot group to determine the most applicable final image for your model or project. The physics create suitable default plots grouped in descriptive plot groups.
- Derived Values and Tables: Derived values define the evaluation of, for example, values of integrals, values of variables in specific locations, and global variables. The evaluation results are stored in Table nodes under **Tables** and displayed in the **Table** window.
- **Export**: You can export a variety of data, images, and animations. See Exporting Data and Images.
- Reports: Create reports as HTML and Microsoft Word documents that contain settings, selections, comments, plots, and other information about the model.
- **Views**: To display the **Views** node under **Results** (for 2D or 3D models), click the **Show** button (👁) and select **Advanced Results Options**. For details, see User-Defined Views.

Results

About the Results Branch

The **Results** branch contains tools and functionality for postprocessing and visualizing of the results. The main **Results** node contains all the nodes that you create for such purposes. The settings window contains the following section:

RESULT SETTINGS

The default is to update all plots automatically when you, for example, recompute the solution, click the plot node, or change the color table for a plot. Clear the **Automatic update of plots** check box to keep the plots unchanged until you explicitly update the plot using the **Plot** button ( ), for example. This can be useful, for example, for large models with complex plots where you do not want to update the plots directly when opening the model or when solving. A blue asterisk in the upper-right corner of the plot node's icon indicates that the plot is not updated ( ).

Changing the Automatic Update of Plots

There are preference settings that you can use to avoid automatic updates of plots when opening or creating models. Open The Preferences Dialog Box and click **Results**. Under **Automatic update of plots** you can set the preferences to update result plots.

- Select the **Disable for new models** check box to always disable automatic update of plots for new models that you create.
- Select the **Disable for models loaded from file (override saved)** check box to always disable automatic update of plots for models that you open from file, such as models in the model libraries. The automatic update of plots is then disabled initially, regardless of the setting in the model.

To activate automatic updates of plots for the current model, select the **Automatic update of plots** check box in the **Result** node's settings window.

THE MAIN RESULTS ANALYSIS AND VISUALIZATION TOOLS

During results analysis and visualization there are these main operation types:

- Data Sets. Data sets contain the source of data for plotting, for example, by indicating a solution and geometry or by transforming another data set.
- Plot Groups and Plots. A plot group is a collection of plots to display simultaneously in the Graphics window. You can enable or disable plots in a plot group to determine the most applicable final image for your model or project. The physics interfaces create suitable default plots grouped in descriptive plot groups. Use a combination of data sets and plot groups to create cross-section plots.
- Derived Values and Tables. Derived values define the evaluation of integrals, maximum and minimum values, values of variables in points, and values of global variables. The evaluation results are stored in tables and displayed in the Table window.
- Exporting Data and Images. Export data, images, and animations from plot groups to files or use a player to visualize dynamic data.
- Reports. Create reports for documenting models as reports with text and images on an HTML format for easy viewing in a web browser or as Word files.

For quick single-click access to the functionality in the **Results** branch, the **Results** toolbar is available for adding plot groups, data sets, data evaluation tools, reports, and other results and visualization tools. When you select a plot group, a plot group contextual toolbar, with the same name as the plot group, appears. From that toolbar you can, for example, add new plots to the plot group and control the window to plot in.

To display the **Views** node under **Results** ( ), click the **Show** button ( ) and select **Advanced Results Options**. This is useful, for example, when 2D axisymmetric revolved plots or 2D cut plane plots for 3D models are created.

- Results Toolbar and Plot Group Contextual Toolbar
- Results in the *COMSOL API Reference Manual*

Common Results Node Settings

Under Results there are common sections on the settings windows. Table 20-1 provides cross references to the information relevant to these nodes, although the same section can also be available for other nodes throughout COMSOL. For the **Coloring and Style** section, see Table 20-2.

COMMON BUTTONS ON THE SETTINGS WINDOWS

The following buttons are available on many of the settings windows and are mostly self-explanatory. These are not explicitly described or explained for every node.

- In general, use the **Move Up** (↑), **Move Down** (↓), or **Delete** ( ) buttons and the fields under tables to edit the table contents. Or right-click a table cell and select **Move Up**, **Move Down**, or **Delete**.
- At any time during plot creation, click the **Plot** button ( ) to preview a data set or plot. Or right-click the node and select **Plot**.
- Click the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons when working with geometric entities in the selection windows and when required.
- Click the **Range** button ( ) to define a range.
- Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.
- Click the **Evaluate** button (=) or right-click the **Derived Values** node and select **Evaluate All** (=) or **Clear and Evaluate All** ( ).

- Going to the Source Node
- About Selecting Geometric Entities
- Entering Ranges and Vector-Valued Expressions

LINKS TO COMMON SETTINGS WINDOW DESCRIPTIONS

TABLE 20-1: DETAILS FOR THE COMMON SETTINGS SECTIONS

SETTINGS WINDOW SECTION	LINK TO MORE INFORMATION
Arrow Positioning	Arrow Positioning
Axis Data	Entering Axis Data for a Data Set
Color (3D plot group, Far Field plots)	Expressions and Predefined Quantities

TABLE 20-1: DETAILS FOR THE COMMON SETTINGS SECTIONS

SETTINGS WINDOW SECTION	LINK TO MORE INFORMATION
Coloring and Style	See Table 20-2 below and Defining the Coloring and Style
Data (for Plots)	Selecting a Data Set for Plots
Data (for Derived Values and Export)	Vector Inputs for Parametric Solver and Parametric Sweep Studies
Data Series Operation	Data Series Operation Settings for a Derived Value
Data for Parametric Solver and Parametric Sweep studies	Vector Inputs for Parametric Solver and Parametric Sweep Studies
Element Filter	Defining Element Filters
Expression or Expressions	Expressions and Predefined Quantities
Inherit Style	Inheriting Style Options
Integration Settings	Integration Settings for a Derived Value
Legends	Legends
Levels	Defining the Number of Levels
Node Properties	Node Properties for Reports
Parametric Solver and Parametric Sweep studies	Vector Inputs for Parametric Solver and Parametric Sweep Studies
Plane Data	Defining Plane Data for a Data Set
Positioning	Principal Components and Positioning
Principal Components	Principal Components and Positioning
Quality	Entering Quality Settings for Plot Settings Windows
Radius	Expressions and Predefined Quantities and Radius Scale Factor
Range	Defining the Color and Data Ranges
r-Axis Data (polar plots)	Expressions and Predefined Quantities
Selection	About Selecting Geometric Entities
Shrink Elements	Defining Shrinking of Elements
Title	Plot Titles for Plot Groups and Plot Types and Using Special Formats and Symbols in Titles
y-Axis Data (1D plots)	Expressions and Predefined Quantities

COLORING AND STYLE

TABLE 20-2: CROSS REFERENCES FOR THE COMMON COLORING AND STYLE SETTINGS SECTIONS

SECTION	CROSS REFERENCE
Arrow base	Arrow Base
Arrow color	Color
Arrow length	Arrow Length
Arrow type	Arrow Type
Color legend	Color Legend
Coloring	Color Table
Color table	Color Table and Selecting Color Tables
Grid	Grid

TABLE 20-2: CROSS REFERENCES FOR THE COMMON COLORING AND STYLE SETTINGS SECTIONS

SECTION	CROSS REFERENCE
Line color	Color
Line markers	Line Markers or Marker Type
Line style (Line, Color, and Width)	Line Style
Line type	Line Type
Line width	Line Style
Number of arrows	Arrow Placement
Placement	Arrow Placement
Plot along lines when animating	Plot Along Lines When Animating
Point color	Color
Point motion	Point Motion
Point radius	Point Radius
Point style, Point type	Point Style
Radius scale factor	Radius Scale Factor
Range quotient	Range Quotient
Reverse color table	Color Table
Scale factor	Scale Factor
Symmetrize color range	Color Table
Tail and Tail components	Tail and Tail Components
Tail scale factor	Tail Scale Factor
Wireframe	Wireframe

- Plot Groups and Plots
- Derived Values and Tables
- Studies and Solvers
- Entering Ranges and Vector-Valued Expressions

Selecting a Data Set for Plots

Almost every plot type's settings window includes a **Data** section where you select a **Data set** from a list of available and applicable data sets. **From parent** (the default) means that the plot uses the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the data set node that the selection in the list next to the button refers to.

FOR SOME 1D PLOTS

Under **Data** select a **Data set**. Select:

- **From parent** (the default) to use the same data set as the plot group it belongs to.
- A **Solution** data set to visualize a quantity along a geometric edge.
- A **Cut Line** data set to visualize a quantity along the cut line (a cross section).
- A **Parameterized Curve** data set to visualize a quantity along the parameterized curve.

- **None** to not use any of the available data sets.

Data Sets

Entering Axis Data for a Data Set

Revolution 1D and 2D data sets: Specify the revolution axis or point by a method based on the space dimension.

- For a **Revolution 1D** data set, enter a value in the **x** field to specify the revolution point.
- For a **Sector 2D** data set, enter values for both the **x** and **y** coordinates (SI unit: m).
- For a **Mirror 2D**, **Revolution 2D**, or **Sector 3D** data set, from the **Axis entry method** list, select **Two points** to enter the revolution axis by specifying two points or **Point and direction** to specify the axis by specifying one point and a direction vector.
  - If **Two points** is selected, enter coordinates in the **Point 1** and **Point 2** fields for **x** and **y** coordinates (for the 2D data sets), and **x**, **y**, and **z** coordinates for the 3D data set (SI unit: m).
  - If **Point and Direction** is selected, enter **Point** and **Direction** vectors for **x** and **y** coordinates (for 2D data sets), and **x**, **y**, and **z** coordinates for the 3D data set (SI unit: m).

Vector Inputs for Parametric Solver and Parametric Sweep Studies

This information is useful when defining plots or derived values for **Parametric Solver** and **Parametric Sweep** studies. Under the **Data** section in the **Parameter values** section, the associated parameter values are listed.

PLOTS

When setting parameters for parametric sweep studies in the **Results** node, the available settings depend on the problem type. A time-dependent problem, for example, allows you to select both time steps and parameter values. Similarly, an eigenvalue problem contains both eigenvalue and parameter settings. In results nodes, the time and eigenvalue settings are referred to as the inner solutions. Thus, in a graph plot for a parametric eigenvalue solution, for example, the “Solution” setting for the *x*-axis data controls whether you want the inner (that is to say, eigenvalue) or outer (that is to say, parametric) solutions on the *x*-axis.

- Under **Data**, for time-dependent **Parametric Sweep** studies also select an option from the **Select via** list—**Stored output times** or **Interpolated times** (time-dependent models only).

  The time steps cannot be selected directly because the different parametric solutions could have different time steps. Selecting **Stored output times** plots all of them, or select **Interpolated times** to get the same interpolated times for every parameter.
- If **Interpolated times** is selected, enter **Times** or click the **Range** button ( ) to select and define specific times.

DERIVED VALUES

For **Parametric Solver** and **Parametric Sweep** studies, the **Parameter values** section lists the associated parameter values.

For **Parametric Sweep** studies select an option from the **Select via** list: **Stored output times** or **Interpolated times** (time-dependent models only). If the **Parametric Sweep** study has multiple inner solutions, such as time steps, select **Inner solutions** or **Outer solutions** from the **Table rows** list to control which solutions appear in the table rows.

The time steps cannot be selected directly because the different parametric solutions could have different time steps. Selecting **Stored output times** plots all of them, or select **Interpolated times** to get the same interpolated times for every parameter.

- If **Interpolated times** is selected, enter **Times** or click the **Range** button ( ) to select and define specific times.
- When available, from the **Table rows** list, select **Inner solutions** or **Outer solutions**. These options are available when there is a Parametric Sweep problem with dynamic inner solutions (that is to say, time, eigenvalue, or parametric solutions).
  - If **Inner solutions** is selected, when the **Evaluate** button (=) is clicked the results table displays the *dynamic value* (for example, time, eigenvalue, or parametric) solutions in rows.
  - If **Outer solutions** is selected, when the **Evaluate** button (=) is clicked the results table displays the *parameters* in rows.

- Entering Ranges and Vector-Valued Expressions
- Derived Values and Tables

Expressions and Predefined Quantities

When plotting and evaluating results, COMSOL provides a large number of predefined quantities that are specific to the physics in the model as well as general quantities for the geometry, coordinate systems, and mesh.

COMSOL does not limit the results calculations to predefined quantities; you can plot and evaluate any function by entering the corresponding expression. You can combine numbers, parameters, mathematical constants, physical constants, variables, unary operators, and binary operators. The **Expression** field or list is available for most plot types as well as for integration and data display and evaluation. You can enter any expression directly in the field or insert variables from a list of predefined quantities that you open by pressing Ctrl+Space or by clicking the **Insert Expression** (+) button. Any user-defined parameters appear as predefined quantities under **Definitions**.

In the **Expression** sections in the settings windows for plot nodes you can:

- Click the **Replace Expression** ( ) button to select a predefined quantity and replace the contents of the **Expression** field with the corresponding variable.
- Click the **Insert Expression** (+) button to insert the corresponding variable at the current position in the **Expression** field.
- Select a **Unit** from the list. You can select from a predefined number of applicable units for the quantity that the variable represents, but you can also click in the unit's text field and type any compatible unit for that quantity to use a unit that is not in the list (for example, `mi/h` for miles per hour as a unit for a velocity quantity).
- Select the **Description** check box to enter a description (or edit the default).

The predefined quantities that you get access to by clicking one of the above buttons are divided into categories based on where in the model they belong:

- Each physics has its own list of predefined quantities. For multiphysics interfaces, there is typically a common list, with quantities that are not linked to any of the participating physics, and separate lists for each participating physics, with quantities that are specific to each physics.
- A **Definitions** list with variables for coordinate systems and user-defined variables.
- A **Geometry** list with geometry variables such as spatial coordinates and geometric entity indexes.

- A **Mesh** list with mesh variables such as element size and element quality.

> Type a filter text in the text field at the top of the list of predefined expressions to filter the list to only include the quantities that match the filter text. Using a filter text can help you find a predefined quantity of interest without having to search through the full list of quantities.

EVALUATION OF UNDEFINED QUANTITIES

During the evaluation of expressions, by default COMSOL does not report partially undefined quantities, and the program plots a quantity where it is defined. The plot is empty where the plotted data is undefined or "not-a-number" (NaN). If a results quantity is undefined everywhere, an error occurs for all plot types.

ACCESSING OTHER SOLUTIONS THAN THE SELECTED SOLUTION

When you use the names of the dependent variables in a results expression, COMSOL uses the solution associated with the selected parameter value, eigenvalue, or time for a parametric analysis, eigenvalue analysis, or time-dependent analysis, respectively. To access other solutions in the model, use the `with` operator.

PROCESSING SOLUTIONS WITH A STORED LINEARIZATION POINT

If the solution being processed has a stored linearization point (such as for a harmonic perturbation or a small-signal analysis), several options are available for how to evaluate the expression in the **Expression evaluated for** list.

Defining Plane Data for a Data Set

Select a **Plane type**—**Quick** (the default) to specify planes orthogonal to the coordinate axes or **General** to specify general planes. The **Plane type** consists of the sets of planes orthogonal to the coordinate axes applicable for the model geometry—for example, **xy-planes**, **yz-planes**, and **zx-planes** in 3D.

If **Quick** is selected:

- From the **Plane** list, select **xy-planes**, **yz-planes** (the default), **zx-planes**, **yx-planes**, **zy-planes**, or **zx-planes** as the set of planes orthogonal to the coordinate axes applicable for the model geometry. Specify the transverse coordinate by entering the location along the transverse coordinate axis.
- Enter the **x-**, **y-**, or **z-coordinates** in the field based on the **Plane** selection.
  - If **xy-planes** or **yx-planes** is selected, enter the **z-coordinates** (SI unit: m).
  - If **yz-planes or zy-planes** is selected, enter the **x-coordinates** (SI unit: m).
  - If **zx-planes** or **zx-planes** is selected, enter the **y-coordinates** (SI unit: m).

If **General** is selected:

- Select a **Plane entry method**—**Three points** or **Point and normal**. Enter **x**, **y**, and **z** coordinates.

  For the **Mirror 3D** data set, select **Three points** to enter the mirror axis by specifying three points or **Point and normal** to specify the mirror axis by specifying one point and a normal vector.
  - If **Three points** is selected, enter **Point 1**, **Point 2**, and **Point 3** in the **x-**, **y-**, and **z-**coordinate fields (SI unit: m).
  - If **Point and normal** is selected, enter **Point** (SI unit: m) and **Normal** (dimensionless) data in the **x-**, **y-**, and **z-**coordinate fields.

For the **Cut Plane** data set, select the **Additional parallel lines** check box to define multiple planes for plotting or evaluation, for example. Enter **Distances** from the original line in the field, or click the **Range** button ( ) to define a range of distances for additional cut planes. The **Distances** field refers to a direction that is normal to the cut plane.

Plot Titles for Plot Groups and Plot Types

Every plot group and plot type have a **Title** section where the **Title type** is selected and set. The options are **Automatic** (the default), **Custom**, **Manual**, or **None**. An **Automatic** title is generated based on the type of plot or plots selected. Select **Manual** to enter free text in the field or **None** for no title. Select **Custom** to add existing information combined with custom prefix and suffix text to the title as described below.

CUSTOM PLOT TITLES

Solution

Under **Solution** select the check boxes as required.

- Select **Data set** to include details about the data set used for the plot.
- Select **Phase** to include information about the phase (when applicable).
- Select **Solution** to include the details about the solution (the time step or parameter values, for example) for the plot (when applicable).

Type and Data

Under **Type and data** select the check boxes as required.

- Select **Type** to include the plot type in the title.
- Select **Description** to include the variable details.
- Select **Expression** to include the variable expression in the title.
- Select **Unit** to include the variable unit.

User

Under **User** enter text as required.

- Enter text in the **Prefix** field to add free text at the front of any Solution title text string. For example, if all the Data set, Phase, and Solution check boxes are selected, this text is first.

Enter text in the **Suffix** field to add free text at the end of any Solution title text string. For example, if all the Data set, Phase, and Solution check boxes are selected, this text is at the end of this information.

Using Special Formats and Symbols in Titles

Legends

The **Show legends** check box is selected by default to display the plotted expressions to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, these values are also displayed. For 1D point plots, the legend displays the coordinate (or vertex number).

When **Automatic** is selected from the **Legends** list (the default), select or clear the **Expression** and **Description** and check boxes to control what to include in the automatic legends (by default it includes the description only). If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

SUPPORT FOR FORMATTING AND SYMBOLS IN TEXTS

For the titles as well as the x-axis, y-axis, and z-axis labels, you can use formatted strings that include HTML tags, Greek letters, and mathematical symbols. The tables in the following sections provide information about supported format and symbols. In addition to ASCII characters, Greek letters, and the mathematical symbols listed in Table 20-15, COMSOL correctly displays any Unicode-based character that you paste into a title or label field.

HTML TAGS

You can use the following HTML tags in text strings for plot labels and titles:

TABLE 20-3: VALID HTML TAGS

HTML TAG	DESCRIPTION
`<B> </B>`	Enclosed text is rendered using a bold font.
`<I> </I>`	Enclosed text is rendered using an italic font.
`<SUB> </SUB>`	Enclosed text is rendered in subscript with the enclosed text slightly lower than the surrounding text.
`<SUP> </SUP>`	Enclosed text is rendered in superscript with the enclosed text slightly higher than the surrounding text.
`<TT> </TT>`	Enclosed text is rendered using a monospaced font.
`<U> </U>`	Enclosed text is underlined.

GREEK CHARACTERS

The texts in labels and titles in all plots support the following Greek character tags:

TABLE 20-4: VALID GREEK SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\ALPHA	Α	\alpha	α
\BETA	Β	\beta	β
\GAMMA	Γ	\gamma	γ
\DELTA	Δ	\delta	δ
\EPSILON	Ε	\epsilon	ε
\ZETA	Ζ	\zeta	ζ
\ETA	Η	\eta	η
\THETA	Θ	\theta	θ
\IOTA	Ι	\iota	ι
\KAPPA	Κ	\kappa	κ
\LAMBDA	Λ	\lambda	λ
\MU	Μ	\mu	μ
\NU	Ν	\nu	ν
\XI	Ξ	\xi	ξ
\OMICRON	Ο	\omicron	ο
\PI	Π	\pi	π
\RHO	Ρ	\rho	ρ
\SIGMA	Σ	\sigma	σ
\TAU	Τ	\tau	τ
\UPSILON	Υ	\upsilon	υ

TABLE 20-4: VALID GREEK SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\PHI	Φ	\phi	φ
\CHI	Χ	\chi	χ
\PSI	Ψ	\psi	ψ
\OMEGA	Ω	\omega	ω

MATHEMATICAL SYMBOLS

For texts in titles and axis labels, you can use the following mathematical symbols:

TABLE 20-5: VALID MATHEMATICAL SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\approx	≈	\bullet	•
\sim	~	\partial	∂
\prop	∝	\nabla	∇
\neq	≠	\prod	∏
\equiv	≡	\sum	∑
\lequal	≤	\sqrt	√
\gequal	≥	\integral	∫
\ll	>>	\oplus	⊕
\gg	<<	\otimes	⊗
\plusmin	±	\larrow	←
\infinity	∞	\rarrow	→
\deg	°	\lrarrow	↔
\cdot	·	\ldarrow	⇐
\times	×	\rdarrow	⇒

Arrow Positioning

Arrow positioning is available when the plot dimension is the same as the highest dimension available. Arrow positioning is available for 2D arrow surface plots in 2D plot groups but not for 2D arrow surface plots in 3D plot groups.

Under **Arrow Positioning**, and based on space dimension, in the **x grid points**, **y grid points**, and **z grid points** fields (**r grid points** and **z grid points** in 2D axial symmetry) select an **Entry method**—**Number of points** or **Coordinates**:

- If **Number of points** is selected, enter the number of **Points** in each direction (default: 15).
- If **Coordinates** is selected, enter **Coordinates** or click the **Range** button () to select and define specific coordinates.

Entering Ranges and Vector-Valued Expressions

Principal Components and Positioning

These sections are available for the Principal Stress Volume and Principal Stress Surface Plots:

PRINCIPAL COMPONENTS

Under **Principal Components**, select a **Type**—**Principal stress** or **Principal strain**.

- Under **Principal values**, enter information in the **First**, **Second**, and **Third Value** fields. The default are the three principal stresses (`solid.sp1`, `solid.sp2`, and `solid.sp3`, for example, for a Solid Mechanics interface; the prefix is the physics interface identifier), plotted using red, green, and blue arrows, respectively.
- Under **Principal directions**, enter information in the table under **First**, **Second**, and **Third** for the **X**, **Y**, and **Z** coordinate fields. The defaults are the directions (eigenvectors) for the first, second, and third principal stress.

For transient problems, enter a **Time**.

POSITIONING

This section is not available for the 3D Principal Stress Surface plot.

Under **Positioning** select an **Entry method**— **Number of points** or **Coordinates** for the **x grid points**, **y grid points**, and **z grid points**. If **Number of points** is selected, enter the number of **Points** in each direction (the default is 7 for Principal Stress Volume plots and 15 for Principal Stress Surface plots). If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

Defining the Number of Levels

For Contour plots, Contour data sets, and Isosurface data sets, under **Levels** and from the **Entry method** list, select **Number of Levels** or **Levels**.

If **Number of Levels** is selected, enter the total number of levels in the **Total levels** field (the default is 20 for plots and 5 for data sets). Otherwise, enter the values of the contour **Levels** or click the **Range** button ( ) to define specific range of levels.

Selecting Color Tables

For many plot types you can select the color table to use for coloring the surfaces, boundaries, contours, streamlines, slices, and so on. These color tables use 1024 colors each. The best way to compare the color tables is to experiment with the options.

RAINBOW AND RAINBOW LIGHT

Rainbow is the default for plots that support color tables. The color ordering corresponds to the wavelengths of the visible part of the electromagnetic spectrum. It starts at the small-wavelength end with dark blue. The colors range through shades of blue, cyan, green, yellow, and red. The disadvantage of this color table is that people with color vision deficiencies (affecting up to 10% of technical audiences) cannot see distinctions between reds and greens.

RainbowLight is similar but uses lighter colors.

THERMAL, THERMALEQUIDISTANT, AND THERMALLIGHT

Thermal colors range from black through red and yellow to white, corresponding to the colors iron takes as it heats up.

ThermalEquidistant is similar but uses equal distances from black to red, yellow, and white, which means that the black and red regions become larger compared to the Thermal color table.

ThermalLight is similar but uses equal distances from dark red to orange, yellow, and white, which means that the region with the lowest values is red instead of black as it is in the Thermal color table.

CYCLIC

The **Cyclic** color table is useful for displaying periodic functions because it has a sharp color gradient—it varies the hue component of the hue-saturation-value (HSV) color model, keeping the saturation and value constant (equal to 1). The colors begin with red, then pass through yellow, green, cyan, blue, magenta, and finally return to red.

WAVE AND WAVELIGHT

The **Wave** color table is useful for data that naturally have positive and negative attributes in addition to a magnitude. An example of a double-ended scheme, it ranges linearly from blue to light gray, and then linearly from white to red. When the range of the visualized quantity is symmetric around zero, the color red or blue indicates whether the value is positive or negative, and the saturation indicates the magnitude.

People with color vision deficiencies can interpret the **Wave** color table because it does not use red-green-gray distinctions, making it efficient for 99.98% of the population.

WaveLight is similar and ranges linearly from a lighter blue to white (instead of light gray) and then linearly from white to a lighter red.

TRAFFIC AND TRAFFICLIGHT

The **Traffic** color table spans from green through yellow to red. **TrafficLight** is similar but uses lighter colors.

DISCO AND DISCOLIGHT

The **Disco** color table spans from red through magenta and cyan to blue. **DiscoLight** is similar but uses lighter colors.

GRAYSCALE

The **GrayScale** color table uses the linear gray scale from black to white—the easiest palette to understand and order.

Gray scale plots are often easier to use for publication. People can also better perceive structural detail in a gray scale than with color. Use this color table to increase the probability that a plot is interpreted correctly by people with color vision deficiencies.

GRAYPRINT

The **GrayPrint** color table varies linearly from dark gray (RGB: 0.95, 0.95, 0.95) to light gray (RGB: 0.05, 0.05, 0.05). Choose this to overcome two difficulties that the GrayScale color table has when used for printing on paper—it gives the impression of being dominated by dark colors, and white is indistinguishable from the background.

CUSTOM COLOR TABLES

You can also add your own continuous and discrete color tables as files with RGB data

Color Tables in the *COMSOL API Reference Manual*

Defining the Color and Data Ranges

Under **Range**, select the **Manual color range** and **Manual data range** check boxes to manually override the color range and data range, respectively, with values in the **Minimum** and **Maximum** field, or use the sliders to control values.

Defining the Coloring and Style

Depending on the plot type and space dimension, the following options are available and defined under **Coloring and Style**. The items are listed in alphabetical order.

ARROW BASE

Select **Tail** (the default) to position the arrow's tail at the arrow position, or **Head** to position the arrow's head at the arrow position.

ARROW LENGTH

Select an **Arrow length**:

- **Proportional** (the default), so that the length of the arrows is proportional to the magnitude of the quantity they represent.
- **Normalized**, so that all arrows have the same length.
- **Logarithmic**, so that the length of the arrows is proportional to the natural logarithm of the magnitude of the quantity they represent. This makes arrows representing small values relatively larger. The value in the **Range quotient** field (default: 100) determines the ratio between the smallest and largest values in the range of values for the logarithmic arrow length.

ARROW PLACEMENT

Select a **Placement** of the arrows—**Uniform**, **Elements**, or **Uniform anisotropic**.

- Select **Uniform** (the default) for arrows positioned uniformly on the surface
- Select **Elements** for arrows positioned in the mesh elements (that is, more densely placed arrows where the mesh density is high).
- Select **Uniform anisotropic** to position the arrows using an anisotropic density (that is, more arrows in some directions than in others). If **Uniform anisotropic** is selected, use the **x weight**, **y weight**, and (in 3D) **z weight** fields to give weights for the arrow density in the different directions (using positive scalar weights). The default weights are 1 in all directions. A higher value increases the arrow density in the corresponding direction.

Number of Arrows

When **Uniform** or **Uniform anisotropic** is selected as the **Placement**, also specify the **Number of arrows** (default: 200).

ARROW TYPE

Select an **Arrow type**—**Arrow** or **Cone**.

COLOR

For arrows, and unless a **Color Expression** subnode determines the arrow colors, select an arrow **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

For lines, select a **Color**—**Custom**, **Cycle**, **Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White**, or **Yellow**. If you select **Cycle**, it cycles through all the colors. If you select **Custom**, click the **Color** button to select a custom color from the palette. Enter a line **Width** or use the slider to select.

COLOR LEGEND

Color legend is selected by default. Click to clear the check box if required. The legend displays to the right of the plot. Also see Legends.

You can adjust the default precision settings if required. Open The Preferences Dialog Box and click **Graphics and Plot WIndows**. Under **Display format (number of digits)**, in the **Color legend** field, enter an integer between 1 and 15 for the number of digits for the values displayed on the color legend. The default setting is 5 digits.

COLOR TABLE

If the default (**Rainbow** in most plots) is not suitable for the plot, try other options. See Color Tables below for details. In some cases, select a **Coloring**—**Color Table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** or **Custom** to choose a different color.

Reverse Color Table

Select the **Reverse color table** check box to reverse the order of the colors in the color table.

Symmetrize Color Range

Select the **Symmetrize color range** check box to obtain a color range centered around zero. This setting is useful for visualizing wave-like solutions with zero bias.

Selecting Color Tables

GRID

Select a **Grid**—**None** (the default), **Fine**, **Normal**, or **Coarse**. If **Fine**, **Normal**, or **Coarse** are selected, also choose a **Color** for the grid.

INTERPOLATION AND NUMBER OF INTERPOLATED TIMES

Particle trajectories can appear jagged because the output times for the simulation are too few to result in a smooth plot. You can improve the particle trajectories by using a uniform interpolation of the data for the particle trajectories. From the **Interpolation** list, select **None** for no interpolation (the default), or select **Uniform** to use a uniform interpolation of the data using additional interpolated times defined in the **Number of interpolated times** field. The default is 100 interpolated times.

Interpolation of lines is only available for Particle Trajectories plots, which are available for use with Particle data sets created with the Particle Tracing Module.

LINE STYLE

The line styles available depend on the type of plot and the space dimension and include these options.

- **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. If **Cycle** is selected, it cycles through all the options.
- **Line**, **Tube**, or **None**. If **Tube** is selected, enter a **Tube radius expression** for the radius of the tube; click the **Replace Expression** button (▶) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m.

LINE TYPE

Select a **Line type**—**Line** or **Tube**. For 3D Streamline plots, **Ribbon** is also available. Ribbons are an alternative to tubes for visualization of, for example, the vorticity of a flow field.

- If **Tube** is selected, enter a **Tube radius expression** (the radius of the tube); click the **Replace Expression** button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m.
- If **Ribbon** is selected, enter a width for the ribbons in the **Width expression** field; click the Replace Expression button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m. Select the **Width scale factor** check box to enter a user-defined scaling of the ribbons' width in the associated field. By default, the program scales the width automatically.

LINE MARKERS OR MARKER TYPE

Select a **Marker** type—**None**, **Cycle**, **Asterisk**, **Circle**, **Diamond**, **Plus sign**, **Point**, **Square**, **Star**, or **Triangle**.

If a marker is selected (excluding **None**), then from the **Positioning** list select **Interpolated** or **In data points**. For **Interpolated**, enter the **Number** of markers to display (the default is 8; the maximum is 10,000 markers) or use the slider to select. If **In data points** is selected, the markers appear in the data points for the plot (which for a plot of a 1D solution are the mesh nodes).

The line markers are only available for Histogram plots using a continuous function.

LINE WIDTH

Enter a line **Width** or use the slider to select.

PLOT ALONG LINES WHEN ANIMATING

If the plan is to create an **Animation** report, select the **Plot along lines when animating** check box. This is useful for Particle Trajectories, Particle Tracing, and Particle Tracing with Mass plots.

POINT MOTION

Select a **Point motion** to specify what should happen **When particles leaves domain**—**Stick to Boundary** (to plot the points on the boundary at the exit point) or **Disappear** (to not render these points at all).

For static fields, specify the **End time** in the **Advanced** section. It is possible that all particles have left the domain at the selected time. In that case, all points appear at the outflow boundary if **Stick to boundary** is selected, and no points appear if **Disappear** is selected. To make the points appear, specify an earlier end time.

POINT RADIUS

Enter a **Point radius expression**; click the **Replace Expression** button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 mm.

POINT STYLE

Under **Point style**, select a **Type**—**Point**, **None**, or **Comet tail**. If **Point** or **Comet tail** is selected, enter a Point Radius and Radius Scale Factor.

> **Comet tail** is available with the **Particle Tracing**, **Particle Tracing with Mass**, and the **Particle Trajectories** plots and requires a license for the Particle Tracing Module. See Particle Tracing, Particle Tracing with Mass, and Particle Trajectories.
>
> Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity; visually, it is the same as the tail of a comet traveling through space.
>
> These additional settings are available when **Comet tail** is selected—Tail and Tail Components and Tail Scale Factor.

RADIUS SCALE FACTOR

Select the **Radius scale factor** check box to enter a scalar number for the scale factor.

RANGE QUOTIENT

If **Logarithmic** is selected as the arrow length, enter a **Range quotient**, which is the ratio between the maximum arrow length and the arrow length below which no arrow is drawn. The default is 100.

SCALE FACTOR

Enter a **Scale factor** for the arrows using a positive scalar number in the field or by using the associated slider (for scale factors between 0 and 1).

TAIL AND TAIL COMPONENTS

Define the length and direction of the comet tail as a vector expression. For the **Tail expression**, click the **Replace Expression** button (▶) to select a predefined expression to insert a predefined expression into the **Tail, x component**, **Tail, y component**, and **Tail, x component** (for 3D plots) fields. The expressions available are based on the physics interfaces used in the model. The default expressions (typically `pt.nvx`, `p.nvy`, and `pt.nvz`) represent the negative of the particle velocity.

TAIL SCALE FACTOR

Select the **Tail scale factor** check box to enter a scalar number between 0 and 1 or use the slider to select.

WIREFRAME

To plot only on the visualization mesh, select the **Wireframe** check box. This displays the surface as a triangular grid.

Defining Element Filters

For Mesh and Volume plots you can specify the elements to display under **Element Filter**. Without filtering, the plots display all elements. Using element filters you can highlight elements based on, for example, their mesh quality, size, or location.

To define an element filter, select the **Enable filter** check box and select a **Criterion**—**Logical expression** (the default), **Random** or **Expression**. For Mesh plots, **Worst quality**, **Best quality**, and **Size** are also available, which filters elements with the worst element, best quality, or size, respectively. When you choose **Size**, the fraction that you specify is the fraction with the smallest elements. For example, a fraction of 0.1 plots the smallest 10% of the elements.

- If **Expression** or **Logical expression** is selected, enter an **Expression** in the field. For example, an expression can be `abs(x-y)` to plot a fraction of elements closest to the line $y = x$ (that is, the fraction that you specify is the fraction where the expression evaluates to the smallest values). An example of a logical expression is `(h>0.1)&&(h<0.4)`, which shows the elements with an element size between 0.1 and 0.4 (`h` is the predefined variable for the mesh element size); another example is `x>0`, which plots elements in the right half-plane only.
- If **Random**, **Expression**, **Worst quality**, **Best quality**, or **Size** is selected, specify the **Fraction** of elements (0–1) to show (the default is 1, which means that all elements are included).

Defining Shrinking of Elements

For Mesh and Volume plots, under **Shrink Elements**, enter an **Element scale factor** between 0 and 1 to scale elements in the mesh plot. The default value is 1, which means no shrinking. Using a smaller value shrinks the size of the elements in the plot accordingly. This can be useful for visualizing individual elements and looking at interior elements in a volume plot.

Entering Quality Settings for Plot Settings Windows

Many plots have a **Quality** section where you can select a plot resolution, enforce continuity, and specify the use of accurate derivative recovery. The steps for this section vary slightly based on the plot but are basically as follows.

1 Under **Quality**, select a plot **Resolution**—**Finer**, **Fine**, **Normal**, **Coarse**, **No refinement** or **Custom**. A higher resolution means that elements are split into smaller patches during rendering.

> **Custom** refinement applies to the base data set. The number of elements in the model can therefore increase radically if the plot uses, for example, a revolve data set, since the refinement is applied to the solution data set.

2 To enforce continuity on discontinuous data, under **Quality**, from the **Smoothing** list, select:

- **None**: to plot elements independently.
- **Internal** (the default): to smooth the quantity inside the geometry but no smoothing takes place across borders between domains with different settings.
- **Everywhere**: to apply smoothing to the entire geometry.

The default is to smooth the quantity except across borders between domains, where there is often a sharp transition from one material to another or between different types of physics.

3 Under **Quality**, the **Recover** default is **Off** because the accurate derivative recovery takes processing time. This recovery is a polynomial-preserving recovery that recovers fields with derivatives such as stresses or fluxes with a higher theoretical convergence than smoothing (see Accurate Derivative Recovery).

To edit the default and use accurate derivative recovery, from the **Recover** list select:

- **Within domains**: to perform recovery inside domains.
- **Everywhere**: to apply recovery to all domain boundaries.

ACCURATE DERIVATIVE RECOVERY

Plotting and evaluating stresses or fluxes boils down to evaluating space derivatives of the dependent variables. By default, computing a derivative like `ux` or `uxx` (first and second derivatives of u with respect to x) is done by evaluating the derivative of the shape functions used in the finite element approximation. These values have poorer accuracy than the solution u itself. For example, uxx is identically 0 if u is defined using linear elements. COMSOL Multiphysics evaluates the derivatives (and u itself) using a polynomial-preserving recovery technique by Z. Zhang (see Ref. 1). The recovery is only applied on variables that are discretized using Lagrange shape functions.

The polynomial-preserving recovery is a variant of the superconvergent patch recovery by Zienkiewicz and Zhu that forms a higher-order approximation of the solution on a patch of mesh elements around each mesh vertex. For regular meshes, the convergence rate of the recovered gradient is $O(h^{p+1})$—the same as for the solution itself. Near boundaries the accuracy is not as good, and it might even be worse than without recovery. Results evaluation is about 2–5 times slower when using accurate derivative recovery. For this reason, it is much better to use recovery of stresses or fluxes for surface plots, contour plots, slice plots, and isosurface plots. Use recovery to a lesser extent for volume plots and line plots that plot values near or on the domain boundaries.

By default, the accurate derivative recovery smooths the derivatives within each group of domains with equal settings. Thus, there is no smoothing across material discontinuities. You find the setting for accurate derivative recovery in the plot node's settings windows' **Quality** section. Due to performance reason, the default value for **Recover** list is **Off** (that is, no accurate derivative recovery). Select **Within domains** to smooth the derivatives within each group of domains with equal settings. Select **Everywhere** to smooth the derivatives across the entire geometry.

Reference

1. A. Naga and Z. Zhang, "The Polynomial-Preserving Recovery for Higher Order Finite Element Methods in 2D and 3D," *Discrete and Continuous Dynamical Systems—Series B*, vol. 5, pp. 769–798, 2005.

Inheriting Style Options

All 2D and 3D plot types (except Mesh) have the **Inherit Style** section. Use this functionality to maintain a consistent style between plots.

After there is more than one plot in a plot group, the **Plot** list makes these plots available to select the attributes you want to maintain between plots. For any plot with this section, and if the check box is applicable to the plot type, the **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. The attributes vary based on the plot type and include:

- Arrow scale factor
- Color
- Color and data range
- Deform scale factor
- Height scale factor
- Ribbon width scale factor
- Point radius scale factor
- Tail scale factor
- Tube radius scale factor

The default **Plot** selected is **None**, which means that styles are not inherited for any plots. If you want to inherit a style, add and select a plot type from the **Plot** list. All attributes automatically inherit the style from the selected plot. To prevent a plot attribute from being inherited, clear the check box or select **None** from the **Plot** list.

Integration Settings for a Derived Value

Select a **Method**—**Auto** (the default), **Integration**, or **Summation**.

- **Auto**—the default method, which computes the integral for fields by numerical integration and for reaction forces by summation. Predefined quantities for reaction forces use summation instead of integration. Also, if an expression is specified for the integrand using the `reacf` operator, the automatic setting chooses the summation method.

- **Integration**—the standard numerical integration method (quadrature).
- **Summation**—a summation method is useful for calculating reaction forces. The summation method finds all nodes on the boundary, evaluates the expression in the nodes and sum up the values. Reaction force variables are predefined in the structural mechanics interfaces.

COMSOL Multiphysics automatically determines an appropriate **Integration order** for the expression. To change this, click to clear the check box and enter a number. COMSOL performs the integration elementwise using numeric quadrature of the selected order.

2D Axisymmetric Models

For 2D axisymmetric models, **Volume Integration** and **Surface Integration** are available. COMSOL multiplies the expression (integrand) with `2*pi*r` prior to integration to compute the corresponding volume or surface integral if you do the next step.

- For **Surface Integration**, select the **Compute volume integral** check box.
- For **Line Integration**, select the **Compute surface integral** check box.

Data Series Operation Settings for a Derived Value

These settings are available for all **Derived Value** types. In addition to performing the averaging on each solution in a data series (from a parametric or Time Dependent study) an operation can be applied such as the integral or maximum of the averaged quantity for the data series so that the result is, for example, the integral or maximum of the averaged quantity for each step in the data series. The following operators are available from the **Operation** list. Select:

- **None** (the default) to evaluate the average of the data series
- **Average** to evaluate the average of the average of the data series
- **Maximum** to evaluate the maximum of the average of the data series. If **Maximum** is selected, select an option from the **Find maximum of** list—**Real part** or **Absolute value**.
- **Minimum** to evaluate the minimum of the average of the data series. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.
- **Integral** to evaluate the integral of the average of the data series
- **RMS** (the *root mean square* or *quadratic mean*)
- **Standard deviation**
- **Variance**

The data series operations treat the data points as samples of a continuous function. The average, integral, standard deviation, variance, and RMS values are all computed by performing one or more integrations of this continuous function using the trapezoidal method. The results are consistent with viewing the data series as samples of a continuous function but are not consistent with viewing the data points as discrete samples.

Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings

For details about the solvers and studies, including the availability by module, see Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.

For the plots that incorporate harmonic perturbation, small-signal analysis, or prestressed analysis, additional settings display in a variety of plot windows, in an **Expression evaluated for** list:

EXPRESSION EVALUATED FOR

Select an option from the **Expression evaluated for** list—**Static solution**, **Harmonic perturbation**, **Total instantaneous solution**, **Average for total solution**, **RMS for total solution**, or **Peak value for total solution**. Each option is described below.

See Built-In Operators for information about the operators described in this section.

Static Solution

The expression is evaluated by taking the values of any dependent variables from the linearization point of the solution. This is achieved by wrapping the expression in the `linpoint` operator.

Harmonic Perturbation

If **Harmonic perturbation** is selected, the **Compute differential** check box is also available. If the check box is not selected, the expression is evaluated by taking the values of any dependent variables from the harmonic perturbation part of the solution.

If the **Compute differential** check box is selected (default), the differential of the expression with respect to the perturbation is computed and evaluated at the linearization point. This is achieved by wrapping the expression in the `lindev` operator. For expressions that are linear in the solution, the two options are the same.

Total Instantaneous Solution

The expression is evaluated by adding the linearization point and the harmonic perturbation and taking the real part of this sum. This is achieved by wrapping the expression in the `lintotal` operator. The phase and amplitude of the harmonic perturbation part can be set in the corresponding data set.

Average for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then averaging over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalavg` operator.

RMS for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then taking the RMS over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalrms` operator.

Peak Value for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then taking the maximum over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalpeak` operator.

Node Properties for Reports

When applicable, the **Node Properties** section provides settings for including the node properties for all model nodes. Select the **Include author**, **Include data created**, and **Include version** check boxes to include those properties. For **Comments**, the default setting—**From referenced node**—takes the comments from the node in the model; select **Custom** add other comments or **None** for no comments.

Data Sets

About Data Sets

Data sets refer to the source of data for creating **Plots** and **Reports**. It can be a **Solution**, a **Mesh**, or some transformation or cut plane applied to other data sets—that is, you can create new data sets from other data sets. You add data sets to the **Data Sets** branch ( ) under **Results**.

All plots refer to data sets; the solutions are always available as default data sets. Characteristics of a data set include:

- A visualization mesh
- Mapping to a previous data set (except for solutions and meshes)
- Ability to refer the evaluation to the previous data set

The base data set maps to a solution and geometry or some other source of data. An example of a transformation data set is Revolution 2D, which sweeps a 2D data set into 3D.

Cut Point, Cut Line, Cut Plane, Edge 2D, Edge 3D, and Surface data sets are used in combination with plot groups and Line, Point, and Surface graphs to create cross-sectional plots and plots for data in points, along lines and edges, and on cut planes and surfaces.

ADDING A DATA SET TO THE MODEL BUILDER

In the **Model Builder** under **Results** ( ), right-click **Data Sets** ( ) and select an option from the context menu. Continue defining each data set as described. See Table 20-6 for links to all the types of data sets.

ADDING A SELECTION TO A DATA SET

For the data sets that contain data defined in the model, such as **Solution** data sets ( ) and **Mesh** data sets ( ), you can add a selection so that the results and plots use a subset of the geometry. In the **Selection** subnode ( ), select the geometric entities for which you want to include data in the data set using the **Graphics** window and the settings in the **Selection** subnode's settings window. Select the **Propagate to lower dimensions** check box to make a selection of domains, for example, also include their adjacent boundaries, edges, and points. This can be convenient when using Surface plots on the boundaries of a 3D geometry, for example.

- See Data Set Types for links to the data set descriptions.
- Named Selections
- Meshing
- Introduction to Solvers and Studies

Data Set Types

The following table lists the available data set types, including links to the description of the data set's properties and settings:

TABLE 20-6: DATA SET TYPES

ICON	LINK	DESCRIPTION AND PLOT USE
AV ∫du	Average or Integral	Creates a data set that computes the average of another data set, for example to plot the average.
	Contour	To analyze on 2D contour lines. Use this data set for 2D arrow plots, 2D line plots, and 1D global plots.
3D 2D	Cut Line 2D and Cut Line 3D	To create lines through 2D and 3D geometry to visualize along the line. Use this data set to create 2D or 3D cross-sections line plots.
	Cut Plane	Plots on cut planes are made on 3D data sets and can be visualized in either 2D or 3D plot groups.
3D 2D 1D	Cut Point 1D, Cut Point 2D, and Cut Point 3D	Plot and evaluate a value in a certain point along time or along a parametric solution. Use this data set to create 1D, 2D or 3D cross-sections point plots.
3D 2D	Edge 2D and Edge 3D	Plot and evaluate a value along a boundary (edge) in 2D or an edge in 3D. Use this data set to create plots for data on edges.
f(x)	Function 1D, Function 2D, and Function 3D	Create a data set that can evaluate functions on a domain.
	Isosurface	To visualize isosurfaces in 3D. Use this data set to create arrow surface plots, surface plots, and contour plots. In addition, the contour data set can be applied to the isosurface data set.
	Join	To join the solution data from two Solution data sets. This makes it possible to evaluate and plot the difference between two solutions, for example.
MAX Max MIN Min	Maximum and Minimum	Creates a data set that computes the maximum or minimum of another data set.
	Mesh (Data Set)	To make a mesh available for visualization and results analysis. Use a Mesh node in a plot group to get the actual plot.

TABLE 20-6: DATA SET TYPES

ICON	LINK	DESCRIPTION AND PLOT USE
3D 2D	Mirror 2D and Mirror 3D	To extend a solution defined on one side of an axis to the other side of the axis. Useful for visualization of a solution to an axisymmetric problem or plane of reflection.
3D 2D	Parameterized Curve 2D and Parameterized Curve 3D	To visualize data along a general parameterized curve in 2D or 3D. Use this data set to create a line plot in its original dimension and as a line graph plot in 1D.
	Parameterized Surface	To visualize data on a general parameterized surface. Use this data set to with a surface plot in its original dimension and as any plot type in 2D.
2D 1D	Parametric Extrusion 1D and Parametric Extrusion 2D	To extend another data set by using a parameter, such as time, as a dimension.
	Particle	To visualize particle traces computed by a Particle Tracing interface.
2D 1D	Revolution 1D and Revolution 2D	Use a revolution data set to visualize a 2D or 1D axisymmetric solution in 3D and 2D, respectively.
3D 2D	Sector 2D and Sector 3D	To exploit symmetries in the model to reduce the model size and then, if needed, use rotations and reflections to plot the solution for the entire geometry.
	Solution	To make solutions available for visualization and results analysis. Solvers create Solution data sets automatically.
	Surface	Plot and evaluate a value on surfaces (boundaries) in 3D. Use this data set to create plots for data on surfaces.

Common Results Node Settings

Average or Integral

Select an **Average** (AV) or **Integral** (∫du) data set, found under the **More Data Sets** submenu, to compute the average or integral of another data set for example, to plot the average or integral.

DATA

Select a **Data set** for the data to compute the average or integral. To compute the average or integral for a **Solution** data set, use a **Selection** to define the geometric entity (point, boundary, edge, or domain) to integrate over. Right-click the data set and select **Add Selection**.

SETTINGS

Select an integration **Method**—**Auto** (the default), **Integration**, or **Summation**.

- **Auto**—Computes the integral for fields by numerical integration and for reaction forces by summation. Predefined quantities for reaction forces use summation instead of integration. Also, to specify an expression for the integrand, use the `reacf` operator, and the automatic setting chooses the summation method.
- **Integration**—the standard numerical integration method (quadrature).
- **Summation**—a summation method useful for computing reaction forces. The summation method finds all nodes on the boundary, evaluates the expression in the nodes and sums up the values. Reaction force variables are predefined in the structural mechanics interfaces. This is selected instead of Integration for reaction forces when the automatic selection of integration method is active.

For **Integration order**, COMSOL automatically determines an appropriate integration order for the expression. The default is 4. Click to select the check box to make a different entry in the field. COMSOL then performs the integration elementwise using numeric quadrature of the selected order.

Select a **Geometry level**—**Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. The default means the highest geometry dimension for the data in the data set; typically volumes in 3D, surfaces in 2D, and lines in 1D.

For an integral evaluation data set example see Flow Past a Cylinder: model library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**.

Contour

Use a **Contour** ( ) data set, selected from the **More Data Sets** submenu, for results evaluation on 2D contour lines. Contour lines cannot be parameterized in general, so only 2D arrow plots, 2D line plots, and 1D global plots can be used to visualize the data set. Only 2D data sets can be used.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, and **Levels**.

Cut Line 2D and Cut Line 3D

Use a **Cut Line 2D** ( ) or **Cut Line 3D** ( ) data set to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis nodes available in 1D are available for cut line data sets as well as 3D plots and results analysis nodes for edges. These data sets are also used to create cross-section line plots.

Go to Common Results Node Settings for links to information about the **Data** section.

LINE DATA

Use **Line entry method** to specify the cut line either by **Two points** or by a **Point and direction**.

- If **Two points** is selected, enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D) in the **Point 1** and **Point 2** fields ((SI unit: m). If **Point and direction** is selected, enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D) in the **Point** (SI unit: m) and **Direction** fields.

- The **Bounded by points** check box is selected by default to constrain the line between the defined points.
- Select the **Snap to closest boundary** check box to snap the selected points to the closest boundary in the geometry. Use this option when evaluating a variable that is available on boundaries but not in domains. Otherwise, leave the snapping off (the default setting) to avoid the additional computational cost.

For **Cut Line 2D**, also select the **Additional parallel lines** check box to define multiple lines for plotting or evaluation, for example. Then enter **Distances** from the original line in the field. The **Distances** field refers to a direction that is normal to the cut line and which is rotated 90 degrees counterclockwise relative to the cut line's direction. For example, if the cut line is from (0, 0) to (1, 0), then the distances are along the vector (0, 1) from any point on the cut line.

ADVANCED

Under **Space variable**, you can change the name of the space variable for the cut line's coordinate from its default value (`dn1x`, for example). The space variable name shows in the **Table** window when displaying the data.

For a Cut Line 3D example, see Thin-Film Resistance: model library path **COMSOL_Multiphysics/Electromagnetics/thin_film_resistance**.

Cut Plane

For plots on cut planes, use **Cut Plane** data sets ( ), which are made on 3D data sets and can be visualized in either 2D or 3D plot groups. All plots and results analysis nodes available in 2D are available for cut plane data sets as well as for surfaces in 3D. The cut plane corresponds to an orthogonal 2D coordinate system embedded in 3D. This data set is used to create 3D cross-section surface plots.

Go to Common Results Node Settings for links to information about these sections: **Data** and **Plane Data**.

ADVANCED

Under **Space variables**, you can change the name of the space variables **x** and **y** for the cut plane's coordinates from their default values (`cpl1x` and `cpl1y`, for example). These names appear as column titles for the coordinate columns in the **Table** window when displayed in numerical results from a Cut Plane data set.

Select the **Define normal variables** check box to create variable names for the cut plane's normal vector. Then under **Normal variables**, enter or edit the variable names for the components of the normal vector—**nx**, **ny**, and **nz**. The default names are `cpl1nx`, `cpl1ny`, and `cpl1nz`, respectively.

If you have the:

- AC/DC Module, see Magnetic Lens: model library path **ACDC_Module/Particle_Tracing/magnetic_lens**.
- Batteries & Fuel Cells Module, see Mass Transport Analysis of a High Temperature PEM Fuel Cell: model library path **Batteries_and_Fuel_Cells_Module/PEMFC/ht_pem**.
- CFD Module, see Airflow Over an Ahmed Body: model library path **CFD_Module/Single-Phase_Benchmarks/ahmed_body**.
- Chemical Reaction Engineering Module, see Laminar Static Mixer: model library path **Chemical_Reaction_Engineering_Module/Mixing/laminar_static_mixer**.
- Particle Tracing Module, see Ideal Cloak: model library path **Particle_Tracing_Module/Ray_Tracing/ideal_cloak**.

Cut Point 1D, Cut Point 2D, and Cut Point 3D

Use a **Cut Point 1D** (—▪—), **Cut Point 2D** (▣), or **Cut Point 3D** (▣) data set to plot and evaluate a value in a certain point along time or along a parametric solution and to create cross-sectional point plots. The choice of 1D, 2D, or 3D only controls the type of input the data set accepts. For example, a Cut Point 1D can only be added to 1D data sets, Cut Point 2D can only be added to 2D data sets, and so forth. Any of these can be used to make a point graph plot along time and so forth.

Add a **Cut Point 1D** data set for one point as cross-section data. Add a **Cut Point 2D** or **Cut Point 3D** data set to, for example, plot the value in a certain point along time or along a parametric solution and use the data set in its original dimension. Useful ways to visualize and display data in cut points are through **Point Evaluation** nodes (8.85 e-12) under **Derived Values** and **Point Graph** nodes (∿) under **1D Plot Group** nodes.

Go to Common Results Node Settings for links to information about the **Data** section.

POINT DATA

To specify the point data coordinates using the **Entry method** list:

- Select **Coordinates** (the default) to enter the coordinates. For Cut Point 1D, enter an **x** coordinate, for Cut Point 2D, enter **x** and **y** coordinates, and for Cut Point 3D enter **x**, **y**, and **z** coordinates.
- Select **From file** to enter or browse to a text file with the cut point data. Enter the path and filename in the **Filename** field, or click **Browse** to browse to the file.
- Select **Grid** to enter grid coordinates for gridded data. For Cut Point 1D enter an **x** coordinate, for Cut Point 2D enter **x** and **y** coordinates, and for Cut Point 3D enter **x**, **y**, and **z** coordinates.
- Select **Regular grid** to specify the number of points in each direction. Enter **Number of x points**, **Number of y points** (Cut Point 2D and 3D), and **Number of z points** (Cut Point 3D only). The default value is 10 points in each direction.

Select the **Snap to closest boundary** check box to snap the selected points to the closest boundary in the geometry. Use this option when evaluating a variable that is available on boundaries but not in domains. Otherwise, leave the snapping off (the default setting) to avoid the additional computational cost.

For a Cut Point 1D example, and if you have the Plasma Module, see Benchmark Model of a Capacitively Coupled Plasma: model library path **Plasma_Module/Capacitively_Coupled_Plasmas/ccp_benchmark**.

For a Cut Point 2D example, see 2D Heat Transfer Benchmark with Convective Cooling: model library path **COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d**.

For a Cut Point 3D example, and if you have the:

- MEMS Module, see Prestressed Micromirror: model library path **MEMS_Module/Actuators/micromirror**.
- Microfluidics Module, see Star-Shaped Microchannel Chip: model library path **Microfluidics_Module/Fluid_Flow/star_chip**.
- RF Module, see Microwave Oven: model library path **RF_Module/Microwave_Heating/microwave_oven**.

Edge 2D and Edge 3D

Use an **Edge 2D** (□) data set or an **Edge 3D** (⧉) data set to plot and evaluate a value along an edge (boundary) in 2D or an edge in 3D. Create a line graph or any other plot type in a **1D Plot Group** to plot data along an edge using one of these data sets. Use an **Edge 2D** data set to plot values on boundaries (edges) in a 2D geometry. Use an **Edge 3D** data set to plot values on edges in a 3D geometry.

Go to Common Results Node Settings for links to information about these sections: **Data** and **Selection**.

Function 1D, Function 2D, and Function 3D

Adding a **Function 1D**, **Function 2D**, or **Function 3D** node (f(x)) from the **More Data Sets** submenu creates a data set that can evaluate functions on a domain. All functions in the same list as the selected function can be evaluated. The domain is an interval for Function 1D, a rectangle for Function 2D, and a block for Function 3D. The domain does not need to have the same dimension as the number of arguments to the function. To plot the functions, use a line graph in 1D, for example, by pointing to the Function 1D data set in a 1D Plot Group (or similarly for Function 2D and Function 3D).

FUNCTION

Select the function to create a data set for—**None**, **All**, or any of the defined functions in the model. Select **All** to make the Function data set point to all functions in the list rather than to a specific function, which can be useful, for example, to plot several functions together in the same plot group.

PARAMETER BOUNDS

Available fields are based on the function dimension. Enter a **Name**. The **First parameter**, **Second parameter** (Function 2D and 3D), and **Third parameter** (Function 3D only) default names are s, t, and u respectively. For each parameter, enter a **Minimum** lower bound (the defaults are 0) and a **Maximum** upper bound (the defaults are 1) for the first, second, and third dimension of the domain.

RESOLUTION

Enter a **Resolution**. This is the number of points into which each dimension is discretized. The defaults are `1000` (Function 1D), `100` (Function 2D), and `30` (Function 3D), and the valid range is between 2 and 1,000,000 points. A high resolution might require significant computational resources.

For an example of a 1D function data set and if you have the AC/DC Module, see Geoelectrics: model library path **ACDC_Module/Other_Industrial_Applications/geoelectrics**.

Isosurface

Use an **Isosurface** () data set, found under the **More Data Sets** submenu, to visualize isosurfaces in 3D. Isosurfaces cannot be parameterized in general so use this data set to create arrow surface plots, surface plots, and contour plots. The contour data set can be applied to the isosurface data set.

- Go to Common Results Node Settings for links to information about the **Data**, **Expression**, and **Levels** sections.
- Data Sets

Join

Use a **Join** () data set to combine the solutions from two **Solution** data sets. The joined data sets makes it possible to compare solutions from two data sets—for example, to evaluate and visualize the difference between two solutions to the same problem using two different meshes in a mesh convergence study or to create the sum over a parametric sweep that contains a few solutions. The Join data set has predefined methods to combine the solutions to get the difference, norm of difference, product, quotient, or sum of the two solution data sets. In addition, two predefined "operators," `data1` and `data2`, correspond to the solution data in the first and second Solution data set, respectively, and make it possible to compare, for example, solutions from different but compatible models or to combine the two solution data sets using another method than the ones that you can select directly.

DATA 1 AND DATA 2

Select a solution data set as `data1` and another solution data set as `data2` from the **Data** lists in the **Data 1** and **Data 2** sections, respectively.

Select which solutions to use from the **Solutions** lists:

- Select **All** (the default) to use all solutions in the data set.
- Select **One** to use one of the available solutions in a time-dependent, parametric, or eigenvalue solution from the list of solutions that appear underneath the **Solutions** list.

The following data set combinations support the option to include all solutions from both data sets:

- Both data sets point to the same solution.
- Both data sets point to two different stationary solutions.
- Both data sets point to two different time-dependent solutions.

COMBINATION

In the **Method** list select a method for combining the solution data sets:

- Select **Difference** (the default) to combine the data sets using a difference: `data1-data2`.
- Select **Norm of difference** to combine the data sets as `abs(data1-data2)`, where `data1` and `data2` are the results of evaluating the expression in the two source data sets. For complex-valued data this corresponds to the Euclidean norm of the difference.
- Select **Explicit** to combine the data sets using an explicit expression with the "operators" `data1` and `data2` in, for example, a plot node's **Expression** field. This can be useful to compare two different dependent variables in two different data sets such as two solutions from two different models using the same geometry. For example, to plot the sum of the variable `u` from the first data set and the variable `v` from the second data set, type `data1(u)+data2(v)`. The scope for `data1` is the model to which the solution data set under **Data 1** belongs, and similarly for `data2`.
- Select **General** to combine the data sets using a general expression in `data1` and `data2` that you type into the **Expression** field that appears. The default, `data1-data2`, is identical to the **Difference** method. This method is useful for combining data sets with similar solution data (from a mesh convergence study, for example) using another method than a difference, product, quotient, or sum.
- Select **Product** to combine the data sets using a product: `data1*data2`.
- Select **Quotient** to combine the data sets using a quotient: `data1/data2`.
- Select **Sum** to combine the data sets using a sum: `data1+data2`.

Maximum and Minimum

Select the **Maximum** evaluation (MAX) or **Minimum** evaluation (MIN) data set, found under the **More Data Sets** submenu, to create a data set that computes the maximum or minimum of another data set.

DATA

Select a **Data set** for the data to compute the maximum or minimum. To compute the maximum or minimum for a **Solution** data set, use a **Selection** to define the geometric entity (point, boundary, edge or domain) to integrate over. Right-click the data set and select **Add Selection**.

SETTINGS

Select a **Geometry level**—**Taken from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. The default means the highest geometry dimension for the data in the data set: typically volumes in 3D, surfaces in 2D, and lines in 1D.

Select an **Element refinement** (default: 2; the element refinement is the number of partitions of an element edge) to adjust the accuracy of the minimum or maximum values.

Mesh (Data Set)

Add a **Mesh** ( ) data set to make a mesh available for visualization and results analysis, typically for mesh-related quantities such as the mesh element quality, which are possible to plot using a mesh data set without computing a solution. Use a **Mesh** node in a plot group to get a plot of the actual mesh.

MESH

Select the **Mesh** to use as the data from the list.

Mirror 2D and Mirror 3D

Use a **Mirror 2D** ( ) data set to extend a solution defined on one side of an axis to the other side of the axis. This can be useful for visualization of a solution to an axisymmetric problem. Use a **Mirror 3D** ( ) data set to extend a solution defined on one side of a plane to the other side of a plane. Both are selected from the **More Data Sets** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Axis Data** (Mirror 2D), and **Plane Data** (Mirror 3D).

ADVANCED

Select the **Define variables** check box to create a **Positive side indicator** variable. The default, `mir1side`, is 1 in the original domain and 0 in the mirror. Use the **Positive side indicator** variable in the **Expression** section of a plot settings window to exclude quantities from the mirror side. Under **Space variables**, enter or edit the variable names for the mirrored coordinate system. Enter or edit the **x**, **y**, and **z** (Mirror 3D only) variable names in the respective fields. The default names are `mir1x`, `mir1y`, and `mir1z`, respectively.

- For a Mirror 3D example, and if you have the Chemical Reaction Engineering Module, see Surface Reactions in a Biosensor: model library path **Chemical_Reaction_Engineering_Module/Surface_Reactions_and_Deposition_Processes/reacting_pillars**.
- For a Mirror 2D example, and if you have the Plasma Module, see DC Glow Discharge: model library path **Plasma_Module/Direct_Current_Discharges/positive_column_2d**.

Parameterized Curve 2D and Parameterized Curve 3D

Use a **Parameterized Curve 2D** ( ) or **Parameterized Curve 3D** ( ) data set to visualize data along a general parameterized curve. Visualize the parameterized curve as a line plot in its original space dimension (2D or 3D) and as a line graph plot in 1D. Select these options from the **More Data Sets** submenu.

Go to Common Results Node Settings for links to information about the **Data** section.

PARAMETER

Enter a **Name** and the **Minimum** and **Maximum** range of the parameter curve.

EXPRESSIONS

Enter functions for the coordinates of the parameter. For Parameterized Curve 2D enter values in the **x** and **y** fields. For Parameterized Curve 3D, enter values in the **x**, **y**, and **z** fields.

If the expressions contain any global parameters, you must do an update of the current solution or re-solve the model before using the data set.

RESOLUTION

Enter the number of subdivisions of the parameter range. The default **Resolution** is 1000, and the valid range is between 2 and 1,000,000 subdivisions. A high resolution might require significant computational resources.

For a Parameterized Curve 3D example, and if you have the Heat Transfer Module, see Radiative Heat Transfer in Finite Cylindrical Media: model library path **Heat_Transfer_Module/Verification_Models/cylinder_participating_media**.

Parameterized Surface

Use a **Parameterized Surface** () data set, selected from the **More Data Sets** submenu, to visualize data on a general parameterized surface. Visualize the parameterized surface as a surface plot in its original dimension and as any plot type in 2D.

Go to Common Results Node Settings for links to information about the **Data** section.

PARAMETERS

Enter a **Name** and a range of the parameter in the **Minimum** and **Maximum** fields for both the **First parameter** and **Second parameter** fields.

EXPRESSIONS

Enter functions for coordinates of the two parameters in the **x**, **y**, and **z** fields.

If the expressions contain any global parameters, you must do an update of the current solution or re-solve the model before using the data set.

RESOLUTION

Enter the number of subdivisions of the parameter ranges. The default **Resolution** is 200, and the valid range is between 2 and 1,000,000 subdivisions. A high resolution might require significant computational resources.

Parametric Extrusion 1D and Parametric Extrusion 2D

Use a **Parametric Extrusion 1D** () or **Parametric Extrusion 2D** () data set to extend another data set by using a parameter, such as time, as a dimension. Select these from the **More Data Sets** submenu.

SETTINGS

Select the **Level scale factor** check box to edit the field. The default is 1. The **Separate levels** check box is selected by default.

Go to Common Results Node Settings for links to information about the **Data** section.

For a 1D Parametric Extrusion example, see The KdV Equation and Solitons: model library path **COMSOL_Multiphysics/Equation-Based_Models/kdv_equation**.

Particle

Particle data sets require the Particle Tracing Module.

Use a **Particle** (⁘) data set, selected from the **More Data Sets** submenu, to visualize particle traces computed by a Particle Tracing Module interface. The Particle data set is automatically created when solving a model containing one of the Particle Tracing Module interfaces if the **Generate default plots** option is selected in the **Study**. Selections can be added to the particle data set, which makes it possible to compute, for example, the number or fraction of particles in a given domain or on a given boundary during results processing. You can visualize the particles using a plot of the particle trajectories in the original dimension, as a Poincaré map, or as a 2D phase portrait.

PARTICLE SOLUTION

Select a **Solution** from the list of solution data. In the **Particle geometry** field, enter the name of the variable for the particle geometry. This corresponds to the hidden geometry on which the particle degrees of freedom are computed. When the Particle data set is generated from the default plots the correct name for the particle geometry is filled in automatically. The particle geometry takes the name `pgeom_<id>` where `<id>` is the interface identifier for the specific Particle Tracing interface.

In the **Position dependent variables** field, enter the names of the dependent variables for the particles' position using commas to separate the variables. Like the particle geometry, the names of the particle degrees of freedom are filled in automatically when the particle data set is generated from the default plots. The names correspond to the particle position degrees of freedom with the Component Identifier prepended. For example, for a 3D Component, with identifier `<comp1>`, and **Dependent Variables** `qx`, `qy`, and `qz` for the interface, the correct expression to enter is `comp1.qx`, `comp1.qy`, and `comp1.qz`.

If you have the AC/DC Module and the Particle Tracing Module, see Magnetic Lens: model library path **ACDC_Module/Particle_Tracing/magnetic_lens**.

Revolution 1D and Revolution 2D

Use a **Revolution 1D** data set to visualize a 1D axisymmetric ( ) solution in 2D. All plot types in 3D or 2D are available for visualization through the revolution data set. Use a **Revolution 2D** data set to visualize a 2D ( ) solution in 3D. All plot types in 3D or 2D are available for visualization through the revolution data set.

Go to Common Results Node Settings for links to information about these sections: **Data** and **Axis Data**.

REVOLUTION LAYERS

From the **Number of layers** list, choose **Normal** (the default), **Fine**, **Coarse**, or **Custom**. The predefined settings adapt the number of layers to the chosen revolution angle, which minimizes the time to plot the revolved geometry for revolution angles that are less than 360 degrees. If **Custom** is selected, enter the number of **Layers** about the revolution axis (default value: 50).

For all choices, enter the **Start angle** (SI unit: deg) for the revolved model. The default is 0 degrees. Enter the **Revolution angle** (SI unit: deg), to revolve the model to see into the geometry in degrees. The default is 360 degrees, that is, a full revolution. Enter negative values to revolve the model in the opposite direction.

An axisymmetric geometry in the *rz*-plane is projected to the *xy*-plane and then rotated about the *y*-axis or to the *xz*-plane and then rotated about the *z*-axis using the start angle and revolution angle.

ADVANCED

For **Revolution 2D**, from the **Map plane to** list, select a plane to map the axisymmetric solution to—**xy-plane** (the default) to map the *rz*-plane to the *xy*-plane and then rotate it about the *y*-axis, or select **xz-plane** to map the *rz*-plane to the *xz*-plane and then rotate it about the *z*-axis.

For **Revolution 1D** and **Revolution 2D**, select the **Define variables** check box to create variable names for the space and angle variables in the revolved geometry. Then under **Space variables**, enter or edit the variable names for the revolved coordinate system. Enter or edit the **x**, **y**, and **z** (Revolution 2D only) variable names in the respective fields. The default names are `rev1x`, `rev1y`, and `rev1z`, respectively.

Under **phi**, enter or edit the variable name for **phi**. Phi is the name of the angle variable in the revolved coordinate system. The default name is `rev1phi`.

For example, the angle variables can be useful to enter Cartesian components of axisymmetric vector fields (such as `ht.tfluxr*cos(rev1phi)` for the *x*-component of a heat flux from a 2D axisymmetric heat transfer model, where `ht.tfluxr` is the radial component of the total heat flux).

Sector 2D and Sector 3D

Use the **Sector 2D** ( ) and **Sector 3D** ( ) data sets, selected from the **More Data Sets** submenu, to make it possible to plot the solution for the full geometry while reducing computation time and memory requirements for complex geometries by exploiting sector symmetries. The geometry must be of a type that can be transformed through the use of rotation or reflection (mirroring). Rotation and reflection are only available when using an even number of sectors. It is also possible to invert the phase (change the sign) when rotating or reflecting.

For example, suppose that there are *N* sectors in a geometry. A Sector data set first evaluates the input expressions in the source data set, creates *N* copies (one for each sector of the geometry), maps and interpolates the data, and transforms the expression components that correspond to vector fields.

Go to Common Results Node Settings for links to information about these sections: **Data** and **Axis Data**.

SYMMETRY

In the **Number of sectors** field enter any integer greater than or equal to 2 (the default is 2) to define the number of sectors in the full geometry. When the **Number of sectors** entered is an even number, select a **Transformation**—**Rotation** (the default) or **Rotation and reflection**. For an uneven number of sectors, only rotation is available and it requires no additional user input.

If **Rotation and reflection** is selected:

- For **Sector 2D**, enter **x** and **y** coordinates for the **Direction of reflection axis**
- For **Sector 3D** enter **x**, **y**, and **z** coordinates for the **Radial direction of reflection plane**.

ADVANCED

Select the **Define variables** check box to create variables for the **Sector number** and the **Space variables**:

- The **Sector number** is an integer number from 0 to the number of sectors minus 1. Each sector has a unique sector number: 0, 1, 2, and 3 for a sector data set that includes four sectors, for example. The default variable for the sector number is `sec1number`, where `sec1` is the data set node's tag.
- The **Space variables** evaluate to the coordinate after the transformation (as opposed to x, y, and z, which evaluate to the coordinates in the underlying data set). The default variable names (the number of which are based on space dimension) are `sec1x` for the x coordinate, `sec1y` for the y coordinate, and `sec1z` for the z coordinate.

When the **Define variables** check box is selected, the input expression is enabled once for each sector, something that increases evaluation time by roughly a factor of the number of sectors (N). If the input expression being evaluated contains any of the space variables, then this evaluated mode is enabled automatically.

Azimuthal Mode Number

When the **Number of sectors** is odd or **Rotation** is selected as the **Transformation** (cyclic symmetry), also enter the **Azimuthal mode number**. The default is 0.

Use the azimuthal mode number to control the source data set and evaluate it with different phases for the different sectors. If the mode number is k, then the phase is shifted with $-2\pi ik/N$ for sector i.

Invert Phase When Rotating and Invert Phase When Reflecting

These check boxes are available in various combinations as follows:

- When **Rotation** is selected as the **Transformation** and the **Number of sectors** is a multiple of 2 (an even number), choose the **Invert phase when rotating** check box to make the phase of the solution change between consecutive sectors.
- When **Rotation and Reflection** is selected as the **Transformation** and the **Number of sectors** is a multiple of 2 (an even number), choose the **Invert phase when reflecting** check box to make the phase change between consecutive sectors. When the **Number of sector**s is a multiple of 4, choose the **Invert phase when rotating** check box to make the phase change between consecutive sectors.

Solution

The **Solution** ( ) data sets make solutions available for visualization and results analysis. Solvers automatically create **Solution** data sets. Right-click to add a **Deformed Configuration** node to the Mesh node.

Solution data sets do not contain the solution but instead refer to a solution stored in a solver node.

- Deformed Configuration
- Remeshing a Deformed Mesh

SOLUTION

- Select a **Solution** to make available for visualization and results analysis.
- If there is more than one Component, select the geometry to perform visualization and results analysis for by selecting the corresponding **Component** from the list.

- Select the **Frame** to evaluate the coordinates in—**Mesh**, **Material**, **Geometry**, or **Spatial**. The default in most cases is the Material frame, and this rarely needs to be changed. This frame selection is used for all results evaluation that uses the solution data set.
- Enter a value for the **Solution at angle (phase)** (SI unit: deg). The default is 0 degrees and evaluates complex-valued expressions by multiplying the solution in the solution data set by a factor of `exp(i*pi*phase/180)` prior to expression evaluation.
- Enter a **Scale factor** (the default is 1; that is, no scaling) to multiply the solution by a real-valued scale factor.

Surface

Use a **Surface** (▢) data set to visualize data on surfaces (boundaries) of a 3D geometry. Refer to this data set to plot and evaluate a value on a surface using a 2D plot group and a **Surface** or **Contour** plot, for example.

Go to Common Results Node Settings for links to information about the **Data** section.

PARAMETERIZATION

Specify how to parameterize the surface. Choose from one of these options in the **x- and y-axes** list to specify what the local x- and y-axes represent:

- **Surface parameters** (the default) uses the parameters of the 3D surface.
- **xy-plane** is the local x- and y-axes representing the global xy-plane.
- **yz-plane** is the local x- and y-axes representing the global yz-plane.
- **zx-plane** is the local x- and y-axes representing the global zx-plane.
- **yx-plane** is the local x- and y-axes representing the global yx-plane.
- **zy-plane** is the local x- and y-axes representing the global zy-plane.
- **xz-plane** is the local x- and y-axes representing the global xz-plane.
- If **Expression** is selected, enter any expression, including the global space coordinates, for example, in the **x-axis** and **y-axis** fields. The default values are **x** and **y**, respectively.

Plot Groups and Plots

About the Plot Groups

A plot group contains one or more plots (for example, combining a surface plot and a streamline plot) using the same data set, such as a solution. You can define plot groups for 1D, 2D, 3D, and polar plots and then create individual or a series of plots in a plot group. Several plot groups of the same type can also be used in a model. Information in the form of data and images can be used to generate a report or exported.

Attributes can also be added as subnodes to a plot to modify the plot's behavior—**Deformation** ( ) attributes deform a plot (to illustrate, for example, structural deformation), **Color Expression** ( ) attributes modify the color of a plot, and filtering to only include parts of the plot is available using the **Filter** attribute ( ). You can also select appropriate color tables for the plots' color expressions based on your audience and what you plan to do with the final analysis.

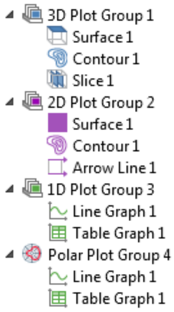
> It is not possible to create plots in a higher dimension than the data set being visualized. For example, a 2D plot group cannot be used to visualize a solution for a 1D model. 1D plot groups can be used for all models.

The physics interfaces create suitable default plots for visualizing the results for the particular physics or application. The default plots appear in plot groups with descriptive names. You can modify and delete these plots and plot groups and add additional plots to existing or new plot groups. To disable the default plots for a study, clear the **Create default plots** check box in the **Study Settings** section in the main **Study** node's settings windows.

Color Coding for Plot Groups and Plot Types

The plot groups and plot types are color coded based on space dimension. See Figure 20-1 for an example.

- All the 1D and polar plots selected from **1D Plot Group** and **Polar Plot Group** nodes are green.
- All the 2D plots selected from the **2D Plot Group** node are pink.
- All the 3D plots selected from the **3D Plot Group** node are blue.



- 3D Plot Group 1
  - Surface 1
  - Contour 1
  - Slice 1
- 2D Plot Group 2
  - Surface 1
  - Contour 1
  - Arrow Line 1
- 1D Plot Group 3
  - Line Graph 1
  - Table Graph 1
- Polar Plot Group 4
  - Line Graph 1
  - Table Graph 1

Figure 20-1: An example of the plot group and plot color coding.

Adding Plots to Plot Groups

Under **Results** ( ) in the **Model Builder**, right-click the **Plot Group** node and select an option from the context menu to plot the graphs listed in Table 20-7. Each plot group can have several plots combined to create a meaningful representation of the data.

When the plot type is defined, click the **Plot** button ( ), or right-click the node and select **Plot**. The plot displays in the window selected from the **Plot window** list. To plot results in another window, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

Also add attributes to 2D and 3D Plot Groups a plot to modify the plot's behavior—**Deformation** ( ) attributes deform a plot, **Color Expression** ( ) attributes modify the color of a plot, and element selection is selected using the **Filter** attribute ( ). A **Height Expression** ( ) can also be added to some 2D plots.

- At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot.
- When you are working with **Functions**, you can also click the **Create Plot** button ( ) to create a customized plot of the function under **Results**, including default plot groups and plots.
- The time-related settings only display on the interfaces for time-dependent models.

- See Plot Types for a summary of all the available plot types, including links to each plot in this guide.
- Results Toolbar and Plot Group Contextual Toolbar
- Creating Cross-Section Plots and Combining Plots

Plot Types

The following table lists the available plot types, including links to the description of the properties and settings.

TABLE 20-7: PLOT TYPES

ICON	NAME AND LINK	DESCRIPTION	PLOT ATTRIBUTES
3D 2D	Arrow Line	Plot a vector quantity as arrows on lines or edges (3D).	Color Expression, Deformation, and Filter
3D 2D	Arrow Surface	Visualize a vector quantity in arrows.	Color Expression, Deformation, and Filter
3D	Arrow Volume	Visualize a vector quantity as arrows in a volume.	Color Expression, Deformation, and Filter
3D 2D	Contour	Visualize a scalar quantity as a contour plot.	Color Expression, Deformation, Filter, and Height Expression (2D only)
3D 2D	Coordinate System Volume, Coordinate System Surface, and Coordinate System Line	Plot coordinate systems for 2D and 3D models. Found on the More Plots submenu.	Deformation (2D surface and 3D volume) Deformation and Filter (2D Line, 3D surface, and 3D line)

TABLE 20-7: PLOT TYPES

ICON	NAME AND LINK	DESCRIPTION	PLOT ATTRIBUTES
3D 2D 1D	Far Field	Plot the value of a global variable. This plot is available for the RF Module and the Acoustics Module.	none
1D	Global	Plot a global scalar quantity as a function of time or a parameter.	Color Expression
1D	Histogram	Plot a histogram showing the distribution of a quantity.	none
3D	Isosurface	Plot a scalar quantity as an isosurface plot.	Color Expression, Deformation, and Filter
1D	Line Graph	Plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line.	Color Expression
3D 2D	Line	Plot a quantity on lines, boundaries (2D), or edges (3D).	Deformation, Filter, and Height Expression (2D only)
Max/Min: Volume Surface 2D Surface 3D Line 2D Line 3D	Max/Min Volume, Max/Min Surface, and Max/Min Line	Plot the maximum and minimum of an expression and the points there they are attained within the geometry. Found on the More Plots submenu.	Deformation and Filter
	Matrix Histogram	Plot a precomputed matrix as a 2D histogram. For rainflow counting, for example, to be able to visualize how the stress amplitudes and mean stresses are distributed. This plot requires the Fatigue Module.	none
	Mesh (Plot)	To display a mesh.	Deformation and Filter

TABLE 20-7: PLOT TYPES

ICON	NAME AND LINK	DESCRIPTION	PLOT ATTRIBUTES
3D	Multislice	Display a scalar quantity on slices in multiple directions inside a 3D domain. Found on the More Plots submenu.	Deformation and Filter
1D	Nyquist	Display a Nyquist plot, which shows the magnitude and phase of a frequency response.	Color Expression
3D 2D	Particle Tracing	Visualize the trajectory of a massless particle subject to a flow field. Found on the More Plots submenu.	Color Expression and Deformation
3D 2D	Particle Tracing with Mass	Make a particle tracing plot considering the particle's mass. Found on the More Plots submenu.	Color Expression and Deformation
3D 2D	Particle Trajectories and Filter Node for Particle Trajectories	Visualize the trajectory of a massless particle subject to a flow field in 2D or 3D. This plot requires the Particle Tracing Module.	Color Expression, Filter, Deformation
2D	Phase Portrait	Visualize particle traces using a phase portrait in 2D. Found under the More Plots submenu. This plot requires the Particle Tracing Module.	Color Expression
3D 2D	Poincaré Map	Visualize particle traces as a Poincaré map in 2D, 2D axial symmetry, and 3D. Found on the More Plots submenu. This plot requires the Particle Tracing Module.	Color Expression
1D	Point Graph	Visualize the value in a point along time or a parameter value. The point can be a point in the geometry or a cut point.	Color Expression
3D	Principal Stress Volume	Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter
3D 2D	Principal Stress Surface	Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter

TABLE 20-7: PLOT TYPES

ICON	NAME AND LINK	DESCRIPTION	PLOT ATTRIBUTES
▦ 2D	Scatter Surface Scatter Surface and Scatter Volume	Scatter surface plots, found on the More Plots submenu, visualize a scalar quantity as scattered spheres or disks. The radius of the spheres can be proportional to the value of a quantity.	None
▦ 3D	Scatter Volume Scatter Surface and Scatter Volume	Scatter volume plots can be used as alternatives to arrow plots for scalar quantities or when to get a feeling for how entities correlate.	None
3D	Slice	Display a scalar quantity on slices inside a 3D domain.	Deformation and Filter
≋ 3D ≋ 2D	Streamline	Plot a vector field as a streamline plot.	Color Expression and Deformation
3D 🟪 2D	Surface	Display a quantity on a 2D domain or a 3D boundary.	Deformation, Filter, and Height Expression (2D plots)
▦ 1D	Table Graph	Table plots display data from a table with one line per output column.	none
🟪 2D	Table Surface	Used with 2D plot groups to visualize the data in a table that represents a matrix of values that are functions of two independent parameters.	Height Expression
🟦 3D	Volume	To display a quantity inside a 3D domain.	Deformation and Filter
	PLOT ATTRIBUTES		**ADD TO THESE PLOTS**
	Color Expression	To add coloring to the shapes defined by a plot.	Arrow (line, surface and volume), contour, isosurface, particle tracing, particle trajectories, principal stress (volume and surface), and streamline
	Deformation	Deform the plot according to a vector quantity, for example, the displacement field in structural mechanics.	All plots except scatter plots.

TABLE 20-7: PLOT TYPES

ICON	NAME AND LINK	DESCRIPTION	PLOT ATTRIBUTES
	Filter and Filter Node for Particle Trajectories	Filter the element selection for a plot.	Arrow (line and surface), contour, coordinate system (line and surface), isosurface, line plot, mesh, principal stress, slice, surface, and volume.
	Height Expression	Height attributes introduce a 3D height on a 2D surface plot.	2D line plot, 2D surface plot, 2D table surface plot
CROSS SECTION PLOTS			
*1D, 2D, and 3D Cross-Section Point Plots		A point cross-section plot makes it easy to view an expression at an arbitrary set of spatial coordinates and results in a line plot.	
*2D Cross-Section Line Plots and 3D Cross-Section Line Plots		Create lines through 2D and 3D geometries to visualize along the line. Also explains how to use the interactive cross-section toolbar buttons.	
*3D Cross-Section Surface Plot		Create planes through a 3D model in a 2D geometry to visualize on the plane. The cut plane corresponds to an orthogonal 2D coordinate system embedded in 3D. Also explains how to use the interactive cross-section toolbar buttons.	
* Cross-section plots are not selected from this menu. Instead, these are created in two steps using data sets and plot groups to generate the plot or use interactive cross-section toolbar buttons.			

The Plot Windows

Plot windows are also graphics windows. COMSOL Multiphysics generates such plot windows for displaying convergence results and to monitor probe values while solving (if your model contains probes). You can also create plot windows manually by choosing **Plot In>New Window** from the context menu for a plot group.

- At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot.
- When you are working with **Functions**, you can also click the **Create Plot** button ( ) to create a customized plot of the function under **Results**, including default plot groups and plots.

This section explains how to add plot windows, specify the window to plot in, and lock plot windows.

ADDING PLOT WINDOWS

The COMSOL Desktop always includes The Graphics Window, which is the default window for all kinds of plots, but you can also add other plot windows for results plots by right-clicking a plot group node and choosing **Plot In>New Window** or by adding a plot window in the **Window Settings** section of the plot group nodes' settings windows.

SPECIFYING THE WINDOW TO PLOT IN

The default for all plot groups is to plot in the **Graphics** window, but you can plot in any other plot window by right-clicking the plot group node and choosing another plot window from the **Plot In** submenu. The **Plot In** submenu also exists on the context menu for each plot type if you want to plot only an individual plot type in a plot group. You can also control where the plots appear and add new plot windows in the **Window Settings** section of the settings windows for the plot group nodes.

To create and update all plots in all plot windows, right-click the **Results** node and select **Plot All** ( ). If more than one plot group use the same plot window, that plot window contains the plots from the last plot group.

LOCKING PLOT WINDOWS

If you want to prevent a plot in a plot window from being overwritten by other plots, you can lock the plot window. To prevent a plot displaying in this window from being overwritten by other plots, click the **Lock Plot Window** button (🔒) on the plot window toolbar. Then any attempt to create a plot in a locked plot window results in a message such as **Window 'Plot 1' is locked** in the **Messages** window. Click the **Lock Plot Window** button (🔒) again to clear the lock. It is not possible to lock the **Graphics** window, which COMSOL Multiphysics uses for general visualization.

- Getting Results While Solving
- Results Toolbar and Plot Group Contextual Toolbar
- The Graphics Window

Creating Cross-Section Plots and Combining Plots

Cross-section plots are created using a combination of data sets and plot groups. Cross-section plots show the values over time, along a parametric solution, or for several eigenvalues. Cross-section plots visualize a quantity as a family of plots on:

- An arbitrary set of points (in 1D, 2D, or 3D)

  A point cross-section plot makes it easy to view an expression at an arbitrary set of spatial coordinates and results in a line plot. See 1D, 2D, and 3D Cross-Section Point Plots.

Expressions and variables that include derivatives of the dependent variables (for example, stresses in a structural analysis) are not available at isolated geometry vertices (points).

- An arbitrary line (in 2D or 3D). See 2D Cross-Section Line Plots and 3D Cross-Section Line Plots.

  Use Cut Line data sets to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis nodes available in 1D are available for Cut Line data sets as well as 3D plots and results analysis nodes for edges.

- Arbitrary planes (in 3D) using a surface plot and cut plane data set. See 3D Cross-Section Surface Plot.

  Use Cut Plane data sets to create planes through a 3D in a 2D geometry to visualize on the plane. All plots and results analysis nodes available in 2D are available for Cut Plane data sets as well as for surfaces in 3D. The cut plane corresponds to an orthogonal (Cartesian) 2D coordinate system embedded in 3D.

A typical cross-section plot uses a Cut Line 2D data set (which you add in the **Data Sets** branch), which defines a straight line (or set of parallel lines) in a 2D geometry, and a Line Graph in a 1D Plot Group, which uses the Cut Line 2D data set as its data input. You can use the same Cut Line data set for multiple cross-section plots of various quantities along the line that the data set defines, and you can create several Cut Line data sets to plot quantities along different lines of interest.

INTERACTIVE CROSS-SECTION LINE AND SURFACE PLOTS

You can also interactively create cross-section line and surface plots using a combination of cross-section toolbar buttons and clicking the geometry. When you use the cross-section toolbar, plot groups and data sets are automatically added and updated in the Model Builder whenever any line or plane is changed. See Creating Interactive 2D Cross-Section Line Plots, Creating Interactive 3D Cross-Section Line Plots, and Creating Interactive 3D Cross-Section Surface Plots.

The following sections give examples how to create cross-section plots.

- Plot Groups and Plots
- Results Toolbar and Plot Group Contextual Toolbar
- See Table 20-7 for links to all the plots.

Plotting and Cross-Section Interactive Toolbar

On the **2D Plot Group** or **3D Plot Group** toolbars, interactive buttons are available based on the plot type. Use these buttons during the creation of cross-section plots or in general while creating plots.

- Plot Group Contextual Toolbar
- Results Toolbar and Plot Group Contextual Toolbar
- See Table 20-7 for links to all the plots.

TABLE 20-8: PLOTTING AND CROSS-SECTION TOOLBAR

ICON	NAME	USE AND RESULT
	Evaluate Along Normal	If you click in a point in the graphics window when a 3D view is shown, then the ray that begins in the point on the screen is intersected with the surfaces of the plot and the color in the first intersection is evaluated to a table with four columns: (x, y, z, color).
	Evaluate	In 2D, if you click on the plot, then the value of the expression that defines the color in that point is evaluated to a table. It contains three columns: (x, y, value).
	First Point for Cut Line	Available for 2D and 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click these buttons to plot a cross-section of data between two points.
	Second Point for Cut Line	
	Cut Line Direction	Available with 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click this button to plot a line perpendicular to a point selected in the Graphics window.

TABLE 20-8: PLOTTING AND CROSS-SECTION TOOLBAR

ICON	NAME	USE AND RESULT
	Cut Line Surface Normal	Available with 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click this button to plot a line in the same way as a domain point probe, with point and direction.
	First Point for Cut Plane Normal	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click these buttons to plot a cross-section of data between the two points along the plane.
	Second Point for Cut Plane Normal	
	Cut Plane Normal	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click this button to plot a plane perpendicular to a point selected in the Graphics window.
	Cut Plane Normal from Surface	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click this button to plot a plane.
	Surface	Click this button to add a Surface plot to a 2D or 3D Plot Group.
	Surface with Height	Click this button to add a surface plot with a height attribute to a 2D Plot Group.
	Arrow Surface	Click this button to add an Arrow Surface plot to a 2D Plot Group.
	Streamline	Click this button to add a Streamline plot to a 2D or 3D Plot Group.
	Line	Click this button to add a Line plot to a 2D or 3D Plot Group.
	Slice	Click this button to add a Slice plot to a 3D Plot Group.
	Isosurface	Click this button to add an Isosurface plot to a 3D Plot Group.
	Volume	Click this button to add a Volume plot to a 3D Plot Group.
	Arrow Volume	Click this button to add an Arrow Volume plot to a 3D Plot Group.

NOTES ABOUT USING THE CROSS-SECTION INTERACTIVE TOOLBAR

The first time any button is clicked on the cross-section toolbar, a data set and a plot group containing either a line graph or surface plot are added to the **Model Builder**. No new data set or plot group is created unless the generated data set or plot groups are deleted or disabled. See below for exceptions. COMSOL chooses the default coordinates for the cross section as a vertical line intersecting the data in the middle.

> !
>
> It is important to ensure the areas of the geometry selected contain data when defining the line or plane. When lines or planes are changed, the coordinates and calculations are automatically updated in the data set and in the final plot.

Deleting and Disabling Data Sets and Plot Groups

The following exceptions apply to the data sets and plot groups that are automatically added using the cross-section toolbar.

- If a plot group is *disabled*, no new data set or plot group is created even if you click one of the interactive buttons. You need to enable the plot group to regenerate the cross-section plot.
- If a plot group is *deleted*, click one of the interactive buttons to regenerate the plot group using the cut plane or cut line data set.
- If a data set is *deleted*, and it is used with a plot group, the plot group is also deleted at the same time. However, if the plot group is using another data set it is not deleted.
- If a data set is *disabled*, the associated plot group is not disabled. However, if you want to plot another cross section, click one of the interactive buttons to create a new data set to use with the plot group.

1D, 2D, and 3D Cross-Section Point Plots

CREATING A 1D CROSS-SECTION PLOT USING A CUT POINT DATA SET

A 1D cross-section point plot visualizes a quantity in one or several points in time, along a parameter range, or for several different eigenvalues.

1. Create a **Cut Point 1D** data set.

   Under **Point Data** enter the **x**-coordinates for the plot. Enter one or several space-separated values or a vector of coordinates, for example, `range(0,10,100)`.
2. Add a **1D Plot Group** ( ). In the settings window, select **Cut Point 1D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-1D Point**.
3. Add a **Point Graph** to this 1D plot group and select **Cut Point 1D** as the **Data set** or **From parent** (the default) to use the same data set as the plot group it belongs to.
4. Continue to define the **Point Graph** as required.
5. Click the **Plot** button ( ), right-click the node and select **Plot**, or press F8.

CREATING A 2D CROSS-SECTION PLOT USING A CUT POINT DATA SET

The 2D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. This example uses the Stresses in a Pulley model from the COMSOL Multiphysics model library.

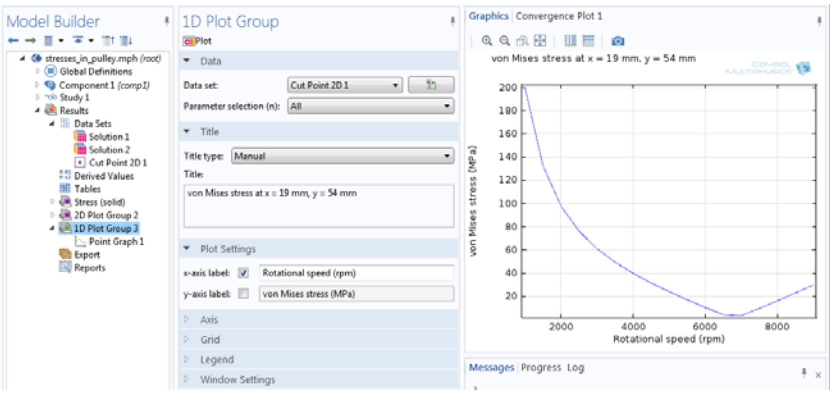
1. Create a **Cut Point 2D** data set. Under **Point Data** enter the *x*- and *y*-coordinate values for the plot. Enter the same number of space-separated values in the **x** and **y** fields. Alternatively, enter a vector of coordinates, for example, `range(0,10,100)`.
2. Add a **1D Plot Group** ( ). In the settings window, select **Cut Point 2D** as the **Data set**.

3 Add a **Point Graph** and select **Cut Point 2D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to.

- The *x*-axis corresponds to time, parameter values, or the eigenvalue number.
- The settings in the **y-axis data** area determine the quantity on the *y*-axis. Select from predefined quantities or enter an expression that contains variables.

4 Continue to define the **Point Graph** as required.

5 Click the **Plot** button ( ) right-click the node and select **Plot**, or press F8.



1D Plot Group

Plot

Data

Data set: Cut Point 2D 1

Parameter selection (n): All

Title

Title type: Manual

Title:

von Mises stress at x = 19 mm, y = 54 mm

Plot Settings

x-axis label: ☑ Rotational speed (rpm)

y-axis label: □ von Mises stress (MPa)

Axis

Grid

Legend

Window Settings

CREATING A 3D CROSS-SECTION PLOT USING A CUT POINT DATA SET

A 3D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. This example uses the “Star-Shaped Microchannel Chip” model from the Microfluidics Module model library.

1 Create a **Cut Point 3D** data set. Under **Point Data** enter the *x*-, *y*-, and *z*-coordinate values for the plot. Enter the same number of space-separated values in the **x**, **y**, and **z** fields. Alternatively, enter a vector of coordinates, for example, `range(0,10,100)`.

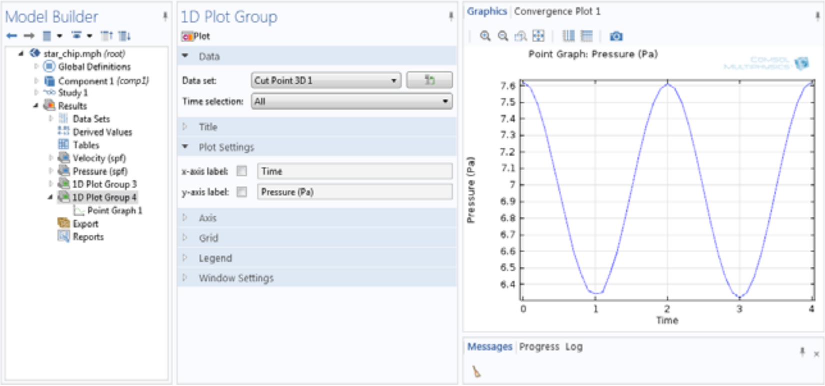
2 Add a **1D Plot Group** ( ). In the settings window, select **Cut Point 3D** as the **Data set**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

3 Add a **Point Graph** and select **Cut Point 3D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

- The *x*-axis corresponds to time, parameter values, or the eigenvalue number.
- The settings in the **y-axis data** area determine the quantity on the *y*-axis. Select from predefined quantities or enter an expression that contains variables.

4 Continue to define the **Point Graph** as required.

5 Click the **Plot** button ( ), right-click the node and select **Plot**, or press F8.



Data set: Cut Point 3D 1

Time selection: All

x-axis label: □ Time

y-axis label: □ Pressure (Pa)

- Plot Groups and Plots
- See Table 20-7 for links to all the plots.

2D Cross-Section Line Plots

The 2D line cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues.

CREATING A 2D CROSS-SECTION PLOT USING A CUT LINE DATA SET

1 Create a **Cut Line 2D** data set.

- Enter the 2D coordinates for the plot. Set the start and end point of the line under **x** and **y**, for Point 1 and Point 2.
- Select the **Additional parallel lines** check box to visualize on a set of parallel lines. Enter the **Distances** from the line as space-separated values.

2 Add a **1D Plot Group** ( ). In the settings window, select **Cut Line 2D** as the **Data set**.

3 Add a **Line Graph** and keep **From parent** to use the same data set as the plot group it belongs to. Settings under **y-Axis Data** and **x-Axis Data** determine the quantity on those axes.

4 Continue to define the **Line Graph** as required.

5 Click the **Plot** button ( ), right-click the node and select **Plot**, or press F8.

CREATING INTERACTIVE 2D CROSS-SECTION LINE PLOTS

1 In the **Model Builder**, click a **2D Plot Group** node to display the buttons available on the **Plot** toolbar.

2 On the **Plot Group** contextual toolbar, click the **First Point for Cut Line** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

3 Click the **Second Point for Cut Line** button ( ). Click an end point on the geometry.

A line connecting the two points is displayed in the **Graphics** window. The first time the cross-section toolbar buttons are clicked, a **Cut Line 2D** data set and a **1D Plot Group** with a **Line Graph** are added to the **Model Builder**.

4 Adjust the cut line as required by clicking the buttons, then click the geometry to change where the first and second point start and end (respectively). The coordinates are updated automatically in the data set and plot group. Click the **1D Plot Group** node to view the updates to the line graph.

5 Continue adjusting the cut line until the line graph representing the points plots the data as required.

- See Table 20-7 for links to all the plots.
- Results Toolbar and Plot Group Contextual Toolbar
- Plot Group Contextual Toolbar

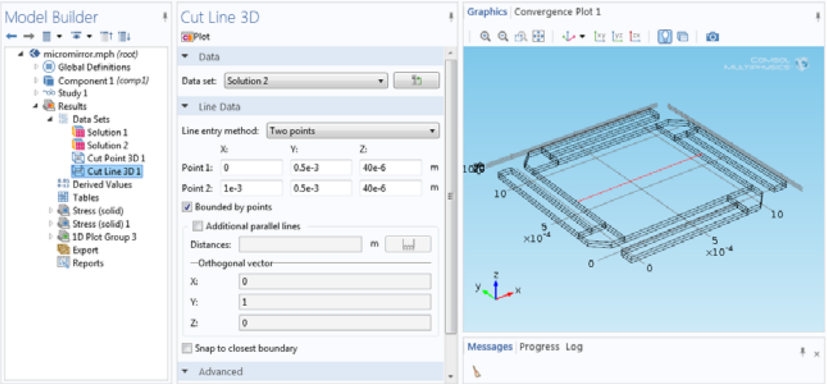
3D Cross-Section Line Plots

A 3D line/extrusion cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues. This example uses the Prestressed Micromirror model from the MEMS Module model library.

CREATING A 3D CROSS-SECTION PLOT USING A CUT LINE DATA SET

1 Create a **Cut Line 3D** data set. Enter the 3D coordinates for the plot. Set the start and end point of the line under **x**, **y**, and **z** for **Point 1** and **Point 2**.

2 Click the **Plot** button ( ).



Cut Line 3D

Plot

Data

Data set: Solution 2

Line Data

Line entry method: Two points

	X:	Y:	Z:	
Point 1:	0	0.5e-3	40e-6	m
Point 2:	1e-3	0.5e-3	40e-6	m

☑ Bounded by points

☐ Additional parallel lines

Distances: m

Orthogonal vector

X: 0

Y: 1

Z: 0

☐ Snap to closest boundary

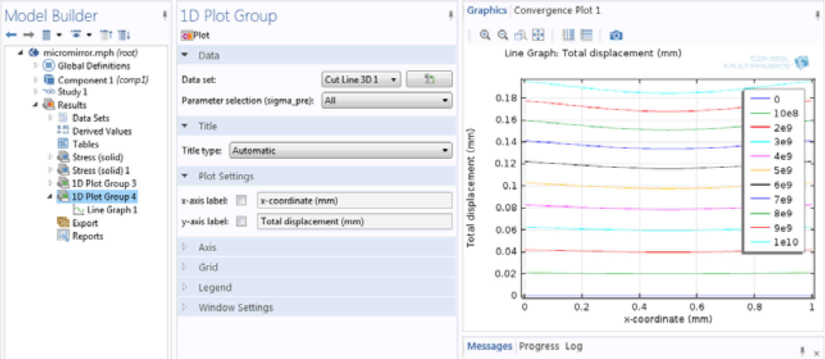
Advanced

3 Add a **1D Plot Group** ( ). In the settings window, select **Cut Line 3D** as the **Data set**.

4 Add a **Line Graph**. For the data set, **From parent** uses the same data set as the plot group it belongs to. Settings under **y-Axis Data** and **x-Axis Data** determine the quantity on those axes.

5 Continue to define the **Line Graph** as required. See Line Graph.

6 Click the **Plot** button ( ) right-click the node and select **Plot**, or press F8.



1D Plot Group

Plot

Data

Data set: Cut Line 3D 1

Parameter selection (sigma_pre): All

Title

Title type: Automatic

Plot Settings

x-axis label: ☐ x-coordinate (mm)

y-axis label: ☐ Total displacement (mm)

Axis

Grid

Legend

Window Settings

CREATING INTERACTIVE 3D CROSS-SECTION LINE PLOTS

1 In the **Model Builder**, click a **3D Plot Group** node to display the buttons available on the **Plot Group** contextual toolbar. On the main toolbar, click the cross-section buttons as required.

The first time the cross-section toolbar buttons are clicked, a **Cut Line 3D** data set and a **1D Plot Group** with a **Line Graph** are added to the **Model Builder**.

Define a Cut Line:

a On the **Plot Group** contextual toolbar, click the **First Point for Cut Line** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

b Click the **Second Point for Cut Line** button ( ). Click an end point on the geometry. A line connecting the two points is created in the **Graphics** window.

c Click either of the buttons, and then click on the geometry to change the start and end points, respectively.

d Click the **1D Plot Group** to view the **Line Graph** based on the selected points.

Define a Cut Line—Direction:

a On the **Plot Group** contextual toolbar, click the **Cut Line Direction** button ( ).

b Click on the geometry to add a line perpendicular to where you clicked. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

c Click the **1D Plot Group** node to view the **Line Graph** based on the selected points.

Define a Cut Line—Surface Normal:

a On the **Plot Group** contextual toolbar, click the **Cut Line Surface Normal** button ( ).

b Click on the geometry to add a line with a point and direction. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

c Click the **1D Plot Group** node to view the **Line Graph** based on the selected points.

2 Adjust the cut line as required by clicking the buttons, then clicking the geometry to change its coordinates. The data set and plot group are updated automatically with the cut line data. Click the **1D Plot Group** node to view the updates to the line graph.

3 Continue adjusting the cut line until the line graph representing the points plots the data as required.

- See Table 20-7 for links to all the plots.
- Results Toolbar and Plot Group Contextual Toolbar
- Plot Group Contextual Toolbar

3D Cross-Section Surface Plot

3D CROSS-SECTION SURFACE PLOT USING A CUT PLANE DATA SET

A 3D surface cross-section plot visualizes a quantity in one or several planes in time, along a parameter range, or for several eigenvalues. This example uses the *Airflow Over an Ahmed Body* model from the CFD Module model library.

1 Create a **Cut Plane 3D** data set.

2 Add a **2D Plot Group** ( ). In the settings window, select **Cut Plane 3D** as the **Data set**.

3 Add a **Surface** plot and keep **From parent** to use the same data set as the plot group it belongs to.

4 Continue to define the **Surface** plot as required.

5 Click the **Plot** button ( ), or right-click the node and select **Plot**.

The plot displays in the window selected in the **Plot window** list. To plot results in another window, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

CREATING INTERACTIVE 3D CROSS-SECTION SURFACE PLOTS

1 In the **Model Builder**, click a **3D Plot Group** node to display the buttons available on the **Plot Group** contextual toolbar. On the main toolbar, click the cross-section buttons as required.

The first time the cross-section toolbar buttons are clicked, a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot are added to the **Model Builder**.

To define a Cut Plane:

a Click the **First Point for Cut Plane Normal** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the cut plane looks like if this first click point is chosen for the surface plot.

b Click the **Second Point for Cut Plane Normal** button ( ). Click an end point on the geometry. The green highlighted areas show you what the cut plane looks like if this second click point is chosen for the surface plot.

c Click either of the buttons and then on the geometry to change the start and end points, respectively.

d Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

To define a Normal Cut Plane:

a Click the **Cut Plane Normal** button ( ).

b Click the geometry to add a plane perpendicular to the click location. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the plot looks like if this normal cut plane is chosen for the surface plot.

c Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

To Define a Normal Cut Plane from Surface:

a Click the **Cut Plane Normal from Surface** button ( ).

b Click the geometry to add a line with a point and direction starting at the click location. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the plot looks like if this normal from surface cut plane is chosen for the surface plot.

c Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

2 Adjust the cut plane as required by clicking the buttons, then clicking the geometry to change the coordinates. The data set and plot group are updated automatically with the cut plane data. Click the **2D Plot Group** node to view the updates to the surface plot.

3 Continue adjusting the cut plane until the surface plot representing the points plots the data as required.

- See Table 20-7 for links to all the plots.
- Results Toolbar and Plot Group Contextual Toolbar
- Plot Group Contextual Toolbar

It is not possible to create plots in a higher dimension than the data set being visualized. For example, a 2D plot group cannot be used to visualize a solution for a 1D model. 1D plot groups can be used for all models.

The time-related settings only display for time-dependent models.

Use a **1D Plot Group** ( ) to plot graphs of, for example, a value that varies with time, or a frequency spectrum, using options to use FFT with a time-dependent solution. You can also use 1D plot groups to create cross-section plots. Normally the plot axes (x-axis and y-axis) use linear scaling but depending on the plotted data it might be beneficial to display the graphs using a log scale on the x-axis or y-axis. From the **Graphics** window's toolbar, click the **x-Axis Log Scale** ( ) and **y-Axis Log Scale** ( ) buttons to use a log scale on the x-axis and y-axis, respectively. Click again to return to a linear scale. For a log scale on an axis, COMSOL tries to show values like $10^{\{\text{integer}\}}$ (for example, 10^1 or 10^{-2}); if it is not possible to show more than three ticks like $10^{\{\text{integer}\}}$, the software instead shows regular numerical values on the axis such as 1, 2, 5,10, 20, 50, 100; and finally, if these regular values are not possible to display, the axes show uniformly distributed regular values.

The **Polar Plot Group** node ( ) creates a graph for *polar plots*: plots of a function in polar coordinates: the radius r and the angle θ. This is useful for visualizing, for example, a radar cross section or other similar polar plots for electromagnetic or acoustic wave models. The available plot types and settings for the **Polar Plot Group** node are similar to those for the 1D Plot Group.

DATA

Select a **Data set**. Depending on the type of data, also specify, for example, the time or frequency selection.

Parametric Sweep Studies

For **Parametric Sweep** studies also select an option from the **Select via** list—**Stored output times** or **Interpolated times**.

- If **Stored output times** is selected, the **Times** section is auto-filled with information from the selected **Data set**. If **Interpolated times** is selected, enter **Times**.

Load Cases

For some solution **Data sets,** you can select the **Load case** to use in the plot group (if you have defined load cases in the model).

Solution Data Sets

For some **Solution Data sets**, select a **Parameter selection (freq)**—**All**, **First**, **Last**, **From list**, or **Manual**.

- If **From list** is selected, select the **Parameter values** from the box that displays.
- If **Manual** is selected, enter **Parameter indices (1-91)** (the actual indices depend on the number of solutions). Or click the **Range** button ( ) to define an **Integer Range**.

TITLE

The **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.

PLOT SETTINGS

Manually enter axis labels by selecting the **x-axis label** and **y-axis label** check boxes.

AXIS

Select the **Manual axis limits** check box to edit the limits already assigned based on the data set. For 1D Plot Groups, this is for the **x minimum**, **x maximum**, **y minimum**, and **y maximum.** For Polar Plot Groups, this is for the **r minimum** and **r maximum**.

For the 1D Plot Group, and as required, select one or all of these check boxes: **Preserve aspect ratio**, **x-axis log scale**, and **y-axis log scale**. When **Preserve aspect ratio** check box is selected, the distances on the *x*-axis and *y*-axis are kept equal.

GRID

Select the **Manual spacing** check box to edit the fields. For 1D Plot Groups, this is for the **x spacing** and **y spacing** fields**.** For Polar Plot Groups, this is for the **r spacing** and θ **spacing** (SI unit: degrees) fields.

For 1D Plot Groups, also specify extra grid points on *x*-axis and *y*-axis in the **Extra x** and **Extra y** fields. For Polar Plot Groups, this is for the **Extra** θ and **Extra r** fields.

LEGEND

Specify the position of the legends for the plots in the plot group. From the **Position** list, select **Upper right** (the default), **Upper left**, **Middle right**, **Middle left**, **Lower right**, or **Lower left**. Legends in all plots in the plot group use this position.

WINDOW SETTINGS

Select a **Plot window**. The **Graphics** window is the default, but any other plot window can be selected, or select **New window** to plot in a new window. Select the **Window title** check box to enter a custom title (except for the **Graphics** window), which is then available in the **Plot window** list for all models. Click the **Add plot window** button (➕) to add a plot window to the list of available windows.

INTERACTIVE (1D PLOT GROUP)

Use a combination of data sets and plots to create a cross-section point plot and cross-section line plot. To add plots to a group, right-click the **1D Plot Group** node to select as many as required. Each plot group can have several plots combined to create a meaningful representation of the data.

You can adjust the default precision settings for the axis labels if required. Open The Preferences Dialog Box and click **Graphics and Plot Windows**. Under **Display format (number of digits)** in the **Graph** field enter an integer between 1 and 15 for the number of digits for the values on the axes in 1D plots and graphs. The default setting is 5 digits.

- Plot Groups and Plots
- Results Toolbar and Plot Group Contextual Toolbar
- See Table 20-7 for a summary of all the available plot types, including links to each plot described in this guide.

2D Plot Group and 3D Plot Group

Use a **2D Plot Group** ( ) to combine one or more 2D plots into and visualize the plots simultaneously. Use a **3D Plot Group** ( ) to combine one of more 3D plots into one to visualize the plots simultaneously.

!	It is not possible to create plots in a higher dimension than the data set being visualized. For example, a 3D plot group cannot visualize a solution for a 2D model. In some cases data sets add dimensions to their parent, for example, Revolve 2D. In this case, the solution is a 2D data set, but the revolved data is a 3D data set, which can be used for 3D plot groups. Similarly, 2D plot groups can be used for 3D solutions when there are, for example, cut planes or other data sets that remove a dimension.

	The time-related settings only display for time-dependent models.

DATA

Select a **Data set**. From the lists below select the solution to use. For example, for **Parametric Sweep** studies select a **Parameter value** as required. For time-dependent problems, select a **Time**.

TITLE

The **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.

PLOT SETTINGS

- Select a **View**. The default is **Automatic**. Other **Views** are defined under **Definitions**.
- (2D only) The **x-axis label** and **y-axis label** check boxes are cleared by default, indicating that empty axis labels are used by default. Select the check boxes to enter labels for the x-axis and the y-axis. This can be useful for scatter plots, for example, where the axes represent other quantities than the x and y directions.
- By default, the plot does not include hidden objects (geometric entities that are hidden in the selected **View**). To include such hidden objects in the plot, select the **Show hidden objects** check box.
- The **Plot data set edges** check box is selected by default. Click to clear if required. Otherwise, select a **Color** (**Black** is the default) or select **Custom** to click the **Color** button and choose a different color from a color palette. Select a **Frame**—**Material** (the default), **Mesh**, **Geometry frame**, or **Spatial**.

COLOR LEGEND

Specify the location of the color legends (color scales) in the plots for this plot group. From the **Position** contains the following positions for the color legends:

- Select **Alternating** to position the first color legend to the right of the plot, the second color legend to the right of the plot, and so on.
- Select **Bottom** to position the color legends horizontally at the bottom of the plot window.
- Select **Left** to position the color legends to the left of the plot.
- Select **Left double** to position the color legends to the left of the plot with two color legends positioned on top of each other (tiled vertically).

- Select **Right** to position the color legends to the right of the plot. This is the default position.
- Select **Right double** to position the color legends to the right of the plot with two color legends positioned on top of each other (tiled vertically).

The default precision for the color legend labels is 5 digits. You can change the precision in the **Preferences** dialog box, using the **Color legend** field under **Display format (number of digits)** on the **Graphics and Plot Windows** page.

WINDOW SETTINGS

- Select a **Plot window**. The **Graphics** window is the default setting, but any other plot window can be selected, or select **New window** to plot in a new window.
- Select the **Window title** check box to enter a custom title (except for the **Graphics** window), which is then available in the **Plot window** list for all models. Click the **Add plot window** button (+) to add a plot window to the list of available windows.

INTERACTIVE

Use a combination of data sets and plots to create a cross-section point plot, cross-section line plot, or cross-section surface plot.

To add plots to a group, right-click the **3D Plot Group** or the **2D Plot Group** node to select as many as required. Each plot group can have several plots combined to create a meaningful representation of the data.

You can adjust the default precision settings for the axis labels if required. Open The Preferences Dialog Box and click **Graphics and Plot Windows**. Under **Display format (number of digits)**:

- In the **2D axis** field, enter an integer between 1 and 15 for the number of digits for the values on the axes in 2D plots. The default setting is 4 digits.
- In the **3D grid** field, enter an integer between 1 and 15 for the number of digits for the values on the axes of the grid in 3D plots. The default setting is 3 digits.

- Plot Groups and Plots
- Results Toolbar and Plot Group Contextual Toolbar
- See Table 20-7 for a summary of all the available plot types, including links to each plot described in this guide.

Arrow Line

Use an **Arrow Line** plot to visualize a vector quantity as arrows on lines using a 2D **Arrow Line** ( ) plot, or lines and edges using a 3D **Arrow Line** ( ) plot. Add Deformation, Color Expression, or Filter subnodes as required.

For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

Arrow Surface

Use an **Arrow Surface** plot to visualize a vector quantity as arrows on a surface using a 2D **Arrow Surface** ( ) or 3D **Arrow Surface** ( ) plot. Add Deformation, Color Expression, or Filter subnodes as required. For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Arrow Volume

Use an **Arrow Volume** ( ) plot visualize a vector quantity as arrows in a 3D volume. Add Deformation, Color Expression, or Filter subnodes as required. For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent. Right-click a **3D Plot Group** to add this plot.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Contour

Use a **Contour** plot to visualize a scalar quantity as a contour in 2D ( ) or 3D ( ) and display the quantity as a set of colored lines. The selected quantity has a constant value on these contour lines, optionally with a 3D height. Add Deformation, Color Expression, Filter, or (2D only) Height Expression subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Levels**, **Quality**, **Inherit Style**, and, except for some details below, **Coloring and Style**.

COLORING AND STYLE

Select a **Contour type**—**Lines** (the default) or **Filled**.

- If **Lines** is selected, you can also select the **Level labels** check box to display line labels on the graph.
- If **Filled** is selected, you can also clear the **Fill surfaces outside of contour levels** check box (selected by default) to not fill the areas of the geometry's surface that are above the highest and below the lowest contour.

If you select to display level labels, specify the precision (number of significant digits) as a positive integer in the **Precision** field (default: 4). Also select a **Label color**: select a predefined color or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Coordinate System Volume, Coordinate System Surface, and Coordinate System Line

Use the **Coordinate System Volume** ( ), **Coordinate System Surface** (2D and 3D), and **Coordinate System Line** ( ) plots to visualize the coordinate systems used in, for example, models of piezoelectric devices, where there can be multiple domains, each using its own set of coordinate systems. Right-click to add a Deformation or Filter (for 2D and 3D Coordinate System Line and 3D Coordinate System Surface plots), as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

COORDINATE SYSTEM

For **Coordinate System Surface** in 2D and **Coordinate System Volume** from the **Source** list you can choose to plot a **Coordinate system** (the default) or a **Matrix variable**:

- For **Coordinate system**, choose one of the available coordinate systems from the **Coordinate system** list. The default is **None** (no coordinate system).
- For **Matrix variable**, click the **Replace Expression** button ( ) to select an available matrix variable from the list, which includes coordinate transforms to and from the added coordinate systems and physical quantities that are tensors, such as the thermal conductivity. The selected variable (for example, `ht.k`), then appears next to **Matrix variable**. Each row of the matrix is plotted as a vector. The first row is plotted in red, the second row in green, and the third row, if any, in blue.

For other coordinate system plots, select another **Coordinate system** to plot. The default is **None** and the list contains any additional coordinate systems that the Component includes.

POSITIONING

This section is available for **Coordinate System Volume** and **Coordinate System Surface** (2D) plots.

In the **x grid points**, **y grid points**, and **z grid points** (3D only) fields select an **Entry method**—**Number of points** or **Coordinates**.

- If **Number of points** is selected, enter the number of **Points** in each direction (the default is 15 for 2D Coordinate System Surface plots and 7 for 3D Coordinate System Volume plots).
- If **Coordinates** is selected, enter **Coordinates** (SI unit: m).

Far Field

> The **Far Field** plots are available with the Acoustics Module or RF Module. For a **3D Plot Group** and **2D Plot Group**, select these plots from the **More Plots** submenu.
>
> Default **Far Field** plots are automatically added to any model that uses far field calculations.

The **Far Field** plots are used to plot the value of a global variable for the far field of an electromagnetic field or acoustic pressure field.

- For the RF Module, the variables are the far-field norm, `normEfar` and `normdBEfar`, or components of the far-field variable `Efar`.
- For the Acoustics Module, the variables are the far-field pressure `ffc1.Lp_pfar` and sound pressure level `ffc1.Lp_pfar`.

The variables are plotted for a selected number of angles on a circle (in 2D) or a sphere (in 3D). The angle interval and the number of angles can be manually specified. Also the circle origin and radius of the circle (2D) or sphere (3D) can be specified. For 3D **Far Field** plots you also specify an expression for the surface color.

The far field plot plots a surface shape by deforming the specified circle or sphere. For each evaluation point on the specified circle or sphere the plot deforms the specified circle or sphere from the evaluation point in the radial direction so that the deformed surface shape distance from the origin becomes equal to the value of the specified expression on the evaluation point on the specified circle or sphere.

The main advantage with the **Far Field** plot, as compared to making a Line Graph, is that the circle or sphere used for defining the plot directions is not part of the geometry for the solution. Thus, the number of plotting directions is decoupled from the discretization of the solution domain.

Add a **Far Field** plot to any plot group (Polar, 1D, 2D, and 3D).

- 1D or Polar plots ( ) for 2D, 2D axisymmetric, or 3D geometry.
- 2D plot ( ) for 2D axisymmetric or 3D geometry.
- 3D plot ( )for 2D axisymmetric or 3D geometry.

> Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Inherit Style**, and **Coloring and Style**. For 1D and Polar plot groups, see the section also for **Legends**. For 3D plot groups, see the list for **Color**.

EVALUATION

1D Plot Group and Polar Plot Group

Under **Angles**, enter the φ **resolution**. The default is 50.

Select a **Restriction**—**None** (the default) or **Manual**. If **Manual** is selected, enter values (SI unit: deg) for φ **start** (the default is 0 degrees) and φ **range** (the default is 360 degrees).

Under **Center**, enter a value for **z-coordinate** (SI unit: mm). The default is 0.

Under **Normal**, enter values for **x**, **y**, and **z**. The defaults for *x* and *y* are 0, and the default for *z* is 1. Enter a **Radius** (SI unit: mm). The default is 1 mm.

2D and 3D Plot Groups

Under **Angles**, enter the θ **resolution**. The default is 10. Enter the φ **resolution**. The default is 20.

Select a **Restriction**—**None** (the default) or **Manual**. If **None** is selected, you can also select the **Compute directivity** check box. If the **Compute directivity** check box is selected, the direction for the strongest radiation and the directivity value display in the **Table** window (see The Table Window and Tables Node).

If **Manual** is selected, enter values (SI unit: deg) for:

- θ **start** (the default is 0 degrees)
- φ **start** (the default is 0 degrees)
- θ **range** (the default is 180 degrees)
- φ **range** (the default is 360 degrees)

Under **Sphere** from the list, select **Unit sphere** (the default) or **Manual**. If **Manual** is selected, enter a value for **z-coordinate** (SI unit: mm). The default is 0. Enter a **Radius** (SI unit: mm). The default is 1 mm.

Global

Use a **Global** ( ) plot to graph a scalar quantity as a function of time or a parameter. Add a Color Expression subnode as required. Right-click a **1D Plot Group** or **Polar Plot Group** to add this plot type.

Go to Common Results Node Settings for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**) **Data**, **Title**, **Coloring and Style**, and **Legends**. Also see the section Expressions and Predefined Quantities.

X-AXIS DATA (1D PLOTS) OR θ ANGLE DATA (POLAR PLOTS)

From the **Parameter** list, select an option for what the *x*-axis represents—**Solution number**, **Frequency spectrum**, **Phase**, or **Expression**. For **Parametric Sweep** studies, and when there are multiple inner solutions, **Parameter value** and **Time** are also available. Select:

- **Solution number** to use the solution numbers as the *x*-axis data (or θ angle data for polar plots) for an eigenvalue solution or for a parametric solution with more than one parameter.
- **Frequency spectrum** to have COMSOL compute the number of frequencies and the frequency range based on the FFT (fast Fourier transform) of the time-dependent solution.
  - To specify these values manually, select the **Number of frequencies** check box and enter a value in the associated field (the default is based on the number of time samples), or
  - Select the **Frequency range** check box and then enter the bounds of the frequency range in the **Minimum** and **Maximum** fields (in Hz). The FFT algorithm uses resampling based on linear interpolation. The *x*-axis shows the frequency (in Hz). By default, the *y*-axis shows the unscaled Fourier coefficients.
  - Select the **Scale** check box to scale the values on the *y*-axis so that their magnitude reflects the magnitude of the original signal. The values then have the same unit as the input data for the FFT. The *y*-axis title includes the unit if all expressions represented on the *y*-axis have the same unit. The scaling makes the magnitude at 0 Hz equal to the bias or DC component of the original signal. For a pure sinusoid the scaled value is the peak magnitude divided by the square root of 2 ($u_{\mathrm{max}}/\sqrt{2}$).
- **Phase** to specify a range of phase angles for the *x*-axis data. The default for the **Phase** is `range(0,0.5,2*pi)` (0–360 degrees in steps of 0.5 rad). Select a **Unit** for the phase angle.

- **Parameter value** to use the *x*-axis data (or θ angle data for polar plots) stored in the solution for a parametric solution with a single parameter.
- **Time** to use time as the *x*-axis data (or θ angle data for polar plots) for a time-dependent solution.

Parametric Sweep Studies

Under **x-Axis Data** (θ **Angle Data** for polar plots), for **Parametric Sweep** studies, and when there are multiple inner solutions, select an option from the **Solutions** list—**Inner** or **Outer**.

- If **Inner** is selected, and for time-dependent studies, the **Times** steps are plotted on the x-axis and one line per parameter is included in the graph (as listed in the **Data>Parameter values** section on this page).
- If **Outer** is selected, one line in the graph is plotted for each inner solution and the **Parameter values** are plotted on the *x*-axis.

One example is a time-dependent problem with a geometric parametric sweep. The time steps are the inner solutions, the parameter sweep the outer solutions.

Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings

Histogram

Use a **Histogram** in 1D ( ) or 2D ( ) to plot a histogram that shows how a quantity is distributed over the geometry (mesh volume). In 1D histograms, the *x*-axis in the histogram represents the values of the quantity (as a number of bins or a range of values), and the *y*-axis represents the count of the total element volume in each interval. You can also view the histogram as a plot showing the area in-between contours or isosurfaces. In 2D histograms the *x*-axis and *y*-axis represent the values of two quantities (as a number of bins or a range of values), and the color surface represents the count of the total element volume in each “bin.” The histogram can be normalized and also displayed as a cumulative plot, and it can appear as a discrete or a continuous function. You can use a histogram with settings that provide a bar chart of, for example, the distribution of values in different ranges. Right-click a **1D Plot Group** or **2D Plot Group** to add this plot. For the 2D Plot Group, select this from the **More Plots** submenu.

- Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Legends**, and **Quality**.
- For a 2D histogram based on a precomputed matrix of data, see Matrix Histogram.

BINS

Select an **Entry method**—**Number of bins** or **Limits**—to define the bins for the histogram’s *x*-axis. Select **Number of bins** (the default) to specify the number of bins (default is 10), or select **Limits** to specify a range of limits (1 2 3 4, for example) for the histogram bins.

For 2D Histogram nodes, these settings are available for the *x*-direction and *y*-direction under **x bins** and **y bins**, respectively.

OUTPUT

Under **Output** specify some properties for the appearance of the histogram. Specify whether to use a continuous or discrete function for the histogram, the normalization, and whether to use a standard or a cumulative histogram.

From the **Function** list, select **Continuous** (the default) to plot the histogram as a continuous function or **Discrete** to plot it as a discrete function (that is, using a constant level in each bin). The discrete version is useful to display the histogram as a bar chart, perhaps with the **Integral** normalization setting so that each bin (bar) shows its relative size.

From the **Normalization** list, select:

- **Integral** to normalize the histogram so that the integral is equal to 1. Use this normalization to get the relative size (percentage) of values in each bin.
- **None** (the default) to show the actual element volume without any normalization.
- **Peak** to normalize the histogram so that the peak values is equal to 1.

Select the **Cumulative** check box to make the histogram cumulative (that is, the value in each bin is the sum of the values for all bins up to the current one).

EVALUATION

Specify the **Space dimension** and the **Geometry level** for the evaluation. By default, the settings are taken from the data set. For a specific model, some space dimensions and geometry levels might not be applicable.

From the **Space dimension** list, select **Take from data set** (the default) or one of the space dimensions **0**, **1**, **2**, or **3**. The default is sufficient except when the data set is, for example, a cut plane, which can be evaluated for space dimensions 2 or 3.

From the **Geometry level** list, select **Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. Using another geometry level than the data set can be useful, for example, for evaluating over the surfaces of a 3D geometry. For solution data sets, **Take from data set** defaults to the highest dimension where there are any mesh elements.

Isosurface

Plot a scalar quantity as an **Isosurface** ( ) plot in 3D. An isosurface plot displays a quantity as a colored set of isosurfaces on which the result has a constant value. The plot can also color isosurfaces based on an independent quantity. You can move the isosurfaces interactively. Add Deformation, Color Expression, or Filter subnodes as required. Right-click a **3D Plot Group** to add this plot type.

Before plotting, select the **Interactive** check box to move the isosurfaces defined in this **Isosurface** node interactively using the slider or by entering a shift in the **Shift** field. A zero shift represents the original position of the isosurfaces.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Levels**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Line Graph

Use a **Line Graph** ( ) to plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line. Make a graph plot of a quantity versus another quantity (for example, time). Add a Color Expression subnode as required. Right-click a **1D Plot Group** or **Polar Plot Group** to add this plot type.

Go to Common Results Node Settings for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**), **Title**, **Coloring and Style**, **Legends**, and **Quality**.

SELECTION (SOLUTION DATA SETS ONLY)

When **Solution** is selected as a **Data Set**, this section displays. Select **Manual** from the **Selection** list to choose geometry directly from the **Graphics** window. Select **All** to add the applicable geometry or any other predefined grouping.

X-AXIS DATA OR θ ANGLE DATA

For **Parametric Sweep** studies, for each pair of outer solution or inner solutions, one line is plotted on the graph. For example, if there are 10 outer solutions and each outer solution has five inner solutions, then 50 lines are drawn. The number of inner solutions can vary between outer solutions.

Select **Arc length** or **Reversed arc length** from the **Parameter** list to visualize along an arc length in the direction of the arc or the reversed direction of the arc, respectively, or select **Expression** to visualize along, for example, a coordinate expression. If **Expression** is selected, go to Expressions and Predefined Quantities.

Line

Use a **Line** plot to display a quantity on lines—that is, boundaries in 2D (□) or edges in 3D (□). Add Deformation, Filter, or Height Expression (2D only) subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plot types.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Matrix Histogram

This plot is available with the Fatigue Module.

Use the **Matrix Histogram** plot in 2D ( ) when you have a precomputed matrix that you want to visualize as a 2D histogram. For example, in a fatigue analysis, you can use it for rainflow counting to be able to visualize how the stress amplitudes and mean stresses are distributed. Using this plot can then help to see how the actual damage is distributed between the different stress levels. If a large fraction of the total damage is caused by loads that occur only a few times, the statistical sample of the loads can be too small to reach any good conclusions.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, and **Coloring and Style**.

EXPRESSION

Click the **Replace Expression** button ( ) to select the matrix variable to use as input for the matrix histogram. The matrix histogram plot uses precomputed matrix variables only.

From the **Unit** list, select any applicable unit for the histogram plot. Select the **Description** check box to customize or enter a description of the plot.

AXES

From the **Unit** list, select an applicable unit for the *x*-axis and *y*-axes of the histogram. The matrix contains data points in the (x, y)-plane where the x- and y-values are stresses. This means that the x- and y-values can have any pressure unit, for example, Pascal. By changing the axes unit, you choose how the x- and y-axes in the plot are interpreted.

Max/Min Volume, Max/Min Surface, and Max/Min Line

Use the **Max/Min Volume** ( ), **Max/Min Surface 2D** ( ), **Max/Min Surface 3D** ( ), **Max/Min Line 2D** ( ), and **Max/Min Line 3D** ( ) plots to plot the maximum and minimum values of an expression and the points there they are attained within the geometry.

When plotting the maximum and minimum value, an associated table appears in the **Table** window (underneath the **Graphics** window if using the default COMSOL Desktop layout). The table contains the maximum and minimum values along with the coordinates for the corresponding locations. The coordinate columns' titles contain the space variable names from the data set, if you use a **Cut Plane** data set, for example. Add a Deformation subnode as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

ADVANCED

Under **Advanced,** define the number of refinements of each mesh element when computing the maximum and minimum by entering a value in the **Element refinement** field (the default is 2). Edit these other settings if required:

- Enter a **Display Precision** for the number of decimals displayed in the labels. The default is 6.
- The **Recover** default is **Off** because recovery takes processing time. To edit the default and use polynomial-preserving recovery and recover fields with derivatives such as stresses or fluxes with a higher theoretical convergence than smoothing, from the **Recover** list, select **Within domains**: to perform recovery inside domains or **Everywhere** to apply recovery to all domain boundaries.
- Select an option from the **Display** list—**Min and max** (the default), **Min**, or **Max**.

Mesh (Plot)

Use a **Mesh** plot ( ) to display a mesh in 2D or 3D. The plot can display the mesh quality or the mesh size. It is possible to plot the mesh without solving a model using a Mesh data set. Add Deformation or Filter subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add this plot type.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Element Filter**, and **Shrink Elements**.

You can also create a Mesh plot by right-clicking the **Mesh** node and selecting **Plot** ( ).

LEVEL

Select a **Level to** display the mesh—**All**, **Volume** (3D only), **Surface**, **Line**, or **Point**. If **Surface** is selected, select the base **Element type** to visualize—**All**, **Triangle**, or **Quad**.

For 3D models and if **Volume** is selected, select the base **Element type** to visualize—**All**, **Tetrahedron**, **Prism**, or **Hex**.

COLOR

Under **Color**, use the following settings to control the coloring of the mesh plot:

- Select an **Element color**—any basic color, **Quality** (the default) to get an element quality plot; **Size** to get a plot of the local mesh size; **Custom** to select a different color from the **Custom color** palette; or **None** to plot with no color.
- Select a **Color table** for the element quality or element size. If the default (**Rainbow**) is not suitable for the plot, try other options.
- Also, if the element color displays the element quality or element size, select the **Color legend** check box (selected by default) to display a color legend next to the plot.
- Also, if the element color displays the element quality or element size, select the **Reverse color table** check box to reverse the colors in the color table so that the color for the maximum value instead indicates the minimum value, and vice versa.
- Select a **Wireframe color**—any basic color, **Custom** to select a different color, or **None** to plot with no color indicating the mesh element boundaries.

Multislice

Use a **Multislice** ( ) plot to display a scalar quantity on slices in multiple directions inside a 3D domain. Add Deformation and Filter subnodes as required. Right-click a **3D Plot Group** to add this plot from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

MULTIPLANE DATA

Under **Multiplane Data**, select an **Entry method**—**Number of planes** or **Coordinates**— for the **x-planes**, **y-planes**, and **z-planes**.

For **Number of planes**, enter the number of planes in the **Planes** field. For **Coordinates**, enter a range of coordinates in the **Coordinates** field.

Nyquist

Use **Nyquist** ( ) to plot a Nyquist plot that shows the magnitude and phase of a frequency response. The plot shows the magnitude as the distance from the origin and the phase as the angle using a curve with the frequency as the parameter. Typical data to use for a Nyquist plot include complex-valued impedance data from a frequency domain study. Add a Color Expression subnode as required. Right-click a **1D Plot Group** or **Polar Plot Group** to add this plot type.

Except where noted below, see Global for all of the settings. For Global Plots, the **Expressions** section is called **y-Axis Data** (or **r-Axis Data** for polar plots), but the instructions are the same.

Under **Coloring and Style**, select the **Show unit circle** check box to include a unit circle in the Nyquist plot.

1D Plot Group and Polar Plot Group

Particle Tracing

This plot type is intended for visualizing a small number of particles on simple geometries. The Particle Tracing Module has vastly superior particle tracing capabilities and should be used for all but the simplest of models.

Use a **Particle Tracing** plot to visualize the trajectory of a massless particle subject to a flow field in 2D ( ) or 3D ( ). Visualize *pathlines* (that is, trajectories of particles released in a flow field), which can be time-dependent or static. For time-dependent flows, also use a snapshot in time of the flow field as a static field. The motion of the particles does not affect the flow field. Add a Color Expression or Deformation subnode as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plot types from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality** (Resolution and Recover only), and **Inherit Style**. See below for sections specific to this plot: **Particle Positioning**, **Release**, **Quality** (ODE solver settings), and **Advanced**.

There is an additional setting under **Coloring and Style** for this plot.

The **Type** of **Point Style** available includes **Comet tail**. Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity—so visually, it is the same as the tail of a comet traveling through space. Go to Common Results Node Settings for the **Comet tail** settings links.

PARTICLE POSITIONING

For 3D models, enter the initial position of particles in the **x**, **y**, and **z** fields. For 2D models, enter the **Positioning** details as described below.

Select a **Positioning**—**Start point controlled** or **Boundary coordinates**. Boundary coordinates are useful, for example, for flow models with one or more inflow boundaries.

- If **Start point controlled** is selected, enter the initial position of particles in the **x** and **y** fields.
- If **Boundary coordinates** is selected, select an item from the **Named selection** list and select an **Entry method**—**Number of points** or **Boundary parameters**.
  - If **Number of points** is selected, enter the number of grid **Points** (the default is 10).
  - If **Boundary parameters** is selected, enter the **Relative coordinates**.

RELEASE

Under **Release**, specify when to **Release particles—Once** (the default), **At intervals**, or **At times**. Select:

- **Once** to release particles once at the first available time, typically at time 0 (zero). To delay the release, select the **Start time** check box and enter a time.
- **At intervals** to release particles at regular intervals starting at the first available time, typically at time 0 (zero). To delay the release, select the **Start time** check box and enter a time. Enter a **Time between releases**. The default is 1.
- **At times** to release particles at an arbitrary time point; enter multiple **Times** to release particles.

QUALITY (ODE SOLVER SETTINGS)

Under **Quality**, also define the **ODE solver settings** as required. Go to ODE Solver Settings—Relative Tolerance, ODE Solver Settings—Absolute Tolerance, and ODE Solver Settings—Step Size for details.

ADVANCED

The **Advanced** section contains settings that do not normally need to be adjusted.

Under **Advanced**, also define these settings as required. Go to Advanced—Termination and Advanced—Instantaneous Flow Field for details.

In the **Termination** section, edit the **Maximum number of steps** and **Edge tolerance.** In the **Instantaneous flow field** section, edit the **Plot static flow field even when time dependent** check box, the **Time variable** default, and the **End time active**.

ODE Solver Settings—Relative Tolerance

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section.

Enter a **Relative tolerance** for the ODE solver. The default is 0.001.

- When solving the 2nd-order ODE $m\ddot{x} = F(t, x, \dot{x})$ for x, the solver first rewrites it as two coupled 1st-order ODEs: one for the position x and one for the velocity $\dot{x}$, each with two components in 2D and three components in 3D.
- The **Relative tolerance** value is the relative error tolerance that the ODE solver uses. It applies to all components of the particle's position and velocity. The solver controls the step size so that the estimated error e in each integration step satisfies

$$\begin{cases} e < \max(\text{atolpos}, \text{rtol} \cdot |x_i|) & \text{(for all components } x_i \text{ of } x\text{)} \\ e < \max(\text{atolvel}, \text{rtol} \cdot |\dot{x}_i|) & \text{(for all components } \dot{x}_i \text{ of } \dot{x}\text{)} \end{cases}$$

where rtol is the relative tolerance specified, atolpos is the absolute tolerance for the particle's position components, and atolvel equals the absolute tolerance for the particle's velocity components.

ODE Solver Settings—Absolute Tolerance

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section. Specify the solver's absolute tolerance. The default is **Automatic**. To enter different values, select **Manual** from the **Absolute tolerance** list and enter a **Position**. The **Position** field can contain a single value—it applies to all components of the position and is the absolute tolerance.

ODE Solver Settings—Step Size

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section. Specify the solver **Step size.** The default is **Automatic**—COMSOL uses the initial value of the acceleration (force divided by mass) and the relative and absolute tolerances to determine the initial time step.

- The automatic maximum step size is 10% of the total simulation time for time-dependent flows as well as for static flow fields where the end time is manually specified in the **Advanced** section (in the **Plot static flow field even when time dependent>End Time** field). For static flow fields where the end time is not set manually, there is no upper limit of the step size. However, in this case, the initial time step is less than or equal to 0.1.

To edit the settings, select **Manual** from the **Step size** list and enter values in the **Initial time step** and **Maximum time step** fields.

- The **Maximum time step** is the longest time step the solver takes. It has higher priority than the **Initial time step**; that is, if an initial step size is set larger than the maximum step size, the solver lowers the initial step size to the maximum step size.

The initial step size, whether entered manually or computed automatically, is not necessarily the first step the solver takes but is a first try. If this step leads to an error such that the tolerances are not met, COMSOL lowers it.

Advanced—Termination

Follow these supplementary instructions for the section found under the **Advanced** section. The **Termination** section contains settings that determine when to end the particle tracing simulation.

- To specify an upper limit of the number of time steps, click to select the **Maximum number of steps** check box and edit the default (1000). The particle simulation ends after this number of steps.
- To specify how close to the geometry boundary the path lines are cut when they exit the geometry, edit the **Edge tolerance** default (0.001). This is a relative tolerance controlling how close to the geometry boundary the pathlines are cut when they exit the geometry. A lower value cuts the line closer to the geometry boundary.

Advanced—Instantaneous Flow Field

Follow these supplementary instructions for the section found under the **Advanced** section. To specify if you want to plot an instantaneous even if the solution is time-dependent, select the **Plot static flow field even when time dependent** check box. This freezes the time selected previously—for example, from a **Plot Group** page in the **Data>Time** list—to the value specified and considers this a static flow field.

- Edit the **Time variable** default (`partt`) if required. Normally it is not necessary to change the default name but the name can be used in expressions as well as for the color when coloring the pathlines according to an expression.
- If required, select the **End time active** check box and enter a value.

Particle Tracing with Mass

This plot type is intended for visualizing a small number of particles on simple geometries. The Particle Tracing Module has superior particle tracing capabilities and should be used for all but the simplest of models.

Use a **Particle Tracing with Mass** plot in 2D ( ) or 3D ( ) to visualize the trajectory of a particle with mass and subject to a flow field. Add a Color Expression or Deformation as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plot types from the **More Plots** submenu.

For particles with mass, COMSOL generates the pathlines by solving the fundamental equation of motion:

$$m\ddot{x} = F(t, x, \dot{x})$$

for the pathline $x(t)$. Here, m is the particle's mass, F equals the force acting upon the particle, and t is time. This is a system of ODEs for x, which COMSOL solves using a pair of Runge-Kutta methods of orders four and five. The solver advances the algorithm with the solution of order five and uses the difference between the order-five and order-four solutions to obtain the local error estimate.

For massless particles, the equation of motion is:

$$\dot{x} = v(t, x)$$

The true formulation of Newton's second law of motion is

$$\frac{d}{dt}(m\dot{x}) = F(t, x, \dot{x})$$

That is, the time derivative of the mass must be considered. The particle-tracing algorithm does not solve this equation. Thus, if an expression is specified for the particle mass that depends on time, the result are incorrect.

Axisymmetric Models

For 2D axisymmetric models, three components for the force are available for particles with mass.

When specifying all three, the algorithm solves for a line in 3D in cylindrical coordinates, but the plot only shows the projection on the axisymmetry plane. In this case, the centripetal force is considered; that is, the algorithm solves the equation

$$\ddot{r} = \frac{F_r}{m} + r\dot{\varphi}^2 \qquad \ddot{\varphi} = \frac{F_\varphi}{rm} - \frac{2\dot{r}\dot{\varphi}}{r} \qquad \ddot{z} = \frac{F_z}{m}$$

where m is the particle mass and (r, φ, z) are the cylindrical coordinates. The variable corresponding to the velocity component in the φ direction (the default name is `partv`) has the dimension length/time, and equals $r\dot{\varphi}$ as $\dot{\varphi}$ has the dimension radians/time.

- Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Coloring and Style**, **Quality** (Resolution and Recover only), and **Inherit Style**.
- See Particle Tracing for **Particle Positioning**, **Release**, **Quality** (ODE solver settings), and **Advanced** settings.
- See Particle Tracing in Fluid Flow for more information about predefined expressions for drag-driven particle movement that are available for particle tracing in fluid-flow models.

There is an additional setting under **Coloring and Style** for this plot.

The **Type** of **Point Style** available includes **Comet tail**. Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity—so visually, it is the same as the tail of a comet traveling through outer space. Go to Common Results Node Settings for the **Comet tail** settings links.

EQUATION OF MOTION

Specify the force acting on the particles. Click the **Replace Expression** (▶) or **Insert Expression** (+) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**—for 2D enter or select **Fx** and **Fy** components of the force, for 3D enter or select **Fx**, **Fy**, and **Fz** components of the force. Enter a **Description** (or edit the default). When some predefined forces are added, there are additional **Parameters** with a **Value** to enter into a table.

MASS AND VELOCITY

Enter the particle **Mass**. Enter the **Initial velocity**—for 2D enter values for the **x component** and **y component**; for 3D enter values for **x component**, **y component**, and **z component**.

QUALITY (ODE SOLVER SETTINGS)

Under **Quality**, also define the **ODE solver settings** as required and described for Particle Tracing. Go to ODE Solver Settings—Relative Tolerance, ODE Solver Settings—Absolute Tolerance, and ODE Solver Settings—Step Size for details.

ADVANCED

Under **Advanced**, define the **Particle velocity variables**. Edit the default variable component names for each particle's velocity. The default names are `partu` (**x component**), `partv` (**y component**), and `partw` (**z component**).

Under **Advanced**, also define these settings as required and described for Particle Tracing. Go to Advanced—Termination and Advanced—Instantaneous Flow Field for details.

Particle Trajectories

This **Particle Trajectories** plot is available with the Particle Tracing Module. However, the plot does not compute the particle trajectories during results processing—the trajectories are computed by one of the physics interfaces in the Particle Tracing Module. The plot can thereby render tens of thousands of particles quickly because the trajectories have already been computed.

Use a **Particle Trajectories** ( ) plot to visualize the trajectory of a massless particle subject to a flow field in 2D or 3D. Add a Color Expression, Deformation, or Filter subnode as required. For the settings in the **Filter** subnode, see Filter Node for Particle Trajectories. Right-click a **2D Plot Group** or **3D Plot Group** to add this plot.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

Filter Node for Particle Trajectories

You can right-click a **Particle Trajectories** plot node ( ) add a **Filter** subnode ( ), which controls the particle types to include and whether to render all particles or a subset of the particles. The Filter node has the following section:

PARTICLE SELECTION

From the **Particles to include** list, select the particle types or subset to include in the particle trajectories plot:

- **All** (the default) to include all particles.
- **Primary** to include primary particles only.
- **Secondary** to include secondary particles only.
- **Logical expression** to include a subset of particles that fulfill the logical expression that you enter in the **Logical expression for inclusion** field. For example, `pt.V>1` only includes particles with a velocity larger than 1, and `x>0` only includes particles in areas where the *x* coordinate is positive.

From the **Particles to render** list, select an option for controlling how many particles to render:

- **All** (the default) to render all particles in the particle tracing simulation.
- **Fraction** to only render a fraction of the particles. You specify the fraction as a number between 0 and 1 in the **Fraction of particles** field. The default fraction is 1; that is, to render all particles.
- **Number** to only render a certain number of particles, which you specify in the **Number of particles** field. The default is to render 100 particles.

Phase Portrait

This plot is available with the Particle Tracing Module.

Use a **Phase Portrait** plot ( ) for 2D and 2D axisymmetric models to visualize large data sets of particle trajectories. The traditional use of a phase portrait is to plot the particle position on the *x*-axis and the particle velocity on the *y*-axis. Each dot in the *xy*-plane represents a particle. By default, the position is taken as the distance from the origin (0, 0, 0) for 3D models. Add a Color Expression as required. Right-click a **2D Plot Group** to add this plot from the **More Plots** submenu.

For 2D and 2D axisymmetric plots, the Phase Portrait view is shared with the model geometry. Therefore, it is necessary to clear the **Plot data set edges** check box found on the **2D Plot Group** or **3D Plot Group** page under **Plot Settings**. Then in the **Graphics** window click the **Zoom extents** button ( ) to see the phase portrait.

EXPRESSION

Select an option from the **x-axis** list—**Position** or **Manual**. If **Manual** is selected, enter an **Expression** (SI unit: m). Select an option from the **y-axis** list—**Speed** or **Manual**. If **Manual** is selected, enter an **Expression** (SI unit: m/s).

The plot is best represented if the magnitude of the *x*-axis data and *y*-axis data are equal. Therefore, it can be useful to normalize the data by selecting **Manual** from the **x-axis** and **y-axis** lists under **Expression** and applying a suitable scaling factor.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

Poincaré Map

This plot is available with the Particle Tracing Module and with 3D models only.

Use a **Poincaré Map** plot to visualize particle trajectories using a *Poincaré map* (sometimes called a *first recurrence map*). Add a Color Expression as required.

The Poincaré map is constructed by first defining a **Cut Plane** ( ) on the **Particle** data set ( ). Then add a **3D Plot Group** or a **2D Plot Group**, depending on the dimension of the particle trajectories, and right-click the plot group node to add these plots from the **More Plots** submenu.

This plot type is useful to visualize the particle trajectories in a plot that represents the position of the particles in a section that is usually transversal to the particle trajectories. The Poincaré map represents the particle trajectories in a space dimension that is one dimension lower than the original particle space.

The Poincaré map parent plot group should point to this cut plane (select a **Cut plane** data set under **Data**). The resulting plot places a dot on the cut plane at the location where a particle crossed the plane. The same particle can cross the cut plane multiple times.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Inherit Style**, and **Coloring and Style**.

Point Graph

Use a **Point Graph** ( ) to visualize the value in a point along time or a parameter value. It can be a point in the geometry or a cut point. Add a Color Expression subnode as required. Right-click a **1D Plot Group** or **Polar Plot Group** to add this plot type.

See Global for these settings: **x-Axis Data** or θ **Angle Data**. Then go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Legends**, **y-Axis** (or **r-Axis**) **Data**, and **Coloring and Style**.

SELECTION (SOLUTION DATA SETS ONLY)

Select **Manual** from the **Selection** list to choose geometry directly from the **Graphics** window. Select **All** to add the applicable geometry or any other predefined grouping.

Principal Stress Volume

Use the **Principal Stress Volume** ( ) to plot the principal stress and principal strain in structural mechanics models. The values of the principal stresses σ_1, σ_2, and σ_3 are the eigenvalues of the stress tensor, ordered such that $\sigma_1 > \sigma_2 > \sigma_3$. The same applies for the principal strains ε_1, ε_2, and ε_3. The plots also show the corresponding eigenvectors using arrows. Add a Deformation, Filter, or Color Expression as required. Right-click a **3D Plot Group** to add this plot from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Principal Components**, **Title**, **Positioning**, **Coloring and Style**, and **Inherit Style**.

Principal Stress Surface

Use the **Principal Stress Surface** plots in 2D ( ) and 3D ( ) to plot the principal stress and principal strain in structural mechanics models. The values of the principal stresses σ_1, σ_2, and σ_3 are the eigenvalues of the stress tensor, ordered such that $\sigma_1 > \sigma_2 > \sigma_3$. The same applies for the principal strains ε_1, ε_2, and ε_3. The plot also shows the corresponding eigenvectors using arrows. Add a Deformation, Filter, or Color Expression as required. Right-click a **2D Plot Group** or **3D Plot Group** to add this plot from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Title**, **Inherit Style**, **Principal Components**, **Positioning**, and **Coloring and Style**.

Scatter Surface and Scatter Volume

Use scatter plots to visualize a scalar quantity as scattered spheres on a 2D **Scatter Surface** ( ) or in a 3D **Scatter Volume** ( ) (as functions of space coordinates or any quantities).

Scatter plots can be used as alternatives to arrow plots for scalar quantities or to represent the correlation between two or more different variables to get a feeling for how quantities correlate. To display a quantity using the color and radius of scattered spheres in the model geometry, use the space coordinates (x, y, and z in 3D) as the expressions for the scatter plot axes.

If you use some other quantities as the variables that determines the scattered spheres' positions on the axes, it is good practice to remove the plotting of the data set's edges (typically the geometry boundaries) by clearing the **Plot data set edges** check box in the main plot group node's settings window. In those cases, the axes in the **Graphics** window no longer represent the space coordinate for the geometry.

The radius and color can both be functions of independent quantities, so a 3D scatter plot can provide information about up to five different quantities as the three axis directions, color, and radius. Right-click a **2D Plot Group** or **3D Plot Group** to add these plot types from the **More Plots** submenu.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Radius**, **Color**, **Coloring and Style**, and **Inherit Style**.

EVALUATION POINTS

Under **Evaluation Points**, select an **Entry method** for the **grid points** coordinates based on space dimension (**x grid points** and **y grid points** for 2D; **r grid points** and **z grid points** for 2D axial symmetry; or **x grid points**, **y grid points**, and **z grid points** for 3D).

The evaluation points are located in a block-shaped (3D) or rectangular (2D) grid where the axes represent the expressions defined in the **Expression** section.

- If **Number of points** is selected, enter the number of **Points** in each direction (the default is 15 for 2D Scatter Surface and 7 for 3D Scatter Volume).
- If **Coordinates** is selected, enter **Coordinates** (SI unit: m).

Slice

Use a **Slice** ( ) to display a scalar quantity on slices inside a 3D domain. Add Deformation and Filter subnodes as required. Right-click a **3D Plot Group** to add this plot.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

PLANE DATA

Under **Plane Data**, select a **Plane Type**—**Quick** (the default) to specify planes orthogonal to the coordinate axes or **General** to specify general planes.

If **Quick** is selected:

- From the **Plane** list, select **xy-planes**, **yz-planes**, or **zx-planes** as the set of planes orthogonal to the coordinate axes applicable for the model geometry.

- Select an **Entry method**—**Number of planes** or **Coordinates**.
  - If **Number of planes** is selected, enter **Planes**.
  - If **Coordinates** is selected, enter the applicable (x, y, or z) grid **Coordinates**. Choose a set of cut plane slices to a coordinate axis, specify the transverse coordinate by entering the location along the transverse coordinate axis in the **Coordinates** field.

If **General** is selected:

- Select an option from the **Plane entry method** list—**Three points** or **Point and normal**.
  - If **Three points** is selected, enter **x**, **y**, or **z** coordinates in the **Point 1**, **Point 2**, and **Point 3** fields.
  - If **Point and normal** is selected, enter **x**, **y**, or **z** coordinates in both the **Point** and **Normal** sections.
- If required, select the **Additional parallel planes** check box and select an **Entry method**— **Number of planes** or **Distances**.
  - If **Number of planes** is selected, enter the number of grid **Planes** (the default is 4).
  - If **Distances** is selected, enter the **Distances** (SI unit: m).

To move the slices interactively, select the **Interactive** check box before plotting. You can then move the slices using the slider or by typing a shift in the **Shift** field. A zero shift represents the original position of the slices.

Streamline

Use a **Streamline** plot in 2D ( ) or 3D ( ) to visualize a vector quantity. A streamline is a curve everywhere tangent to an instantaneous vector field. 3D streamline plot is analogous to the 2D streamline plot except that there is no height data setting and the start point selection is different. Add Deformation or Color Expression subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.

Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality**, and **Inherit Style**.

STREAMLINE POSITIONING

Select one of these options from the **Positioning** list—**On selected boundaries** (the default), **Start point controlled**, **Uniform density**, or **Magnitude controlled**. Then follow one of the methods described:

- Method 1: Specifying the Number of Streamlines and Start Boundaries
- Method 2: Specifying Points by Entering Coordinates
- Method 3: Selecting the Specified Number of Start Points in the Geometry
- Method 4: Creating Streamlines with Uniform Density
- Method 5: Creating Streamlines with Variable Density and Magnitude Controlled

SELECTION

The **Selection** section is available for some data sets when you select **On selected boundaries** from the **Positioning** list under **Streamline Positioning**.

Select the boundaries from which the streamlines start. By selecting in the **Graphics** window and using the tools in the **Selection** section, select the boundaries for the starting positions for the streamlines.

ADVANCED

Define the following advanced streamline settings as required.

Advanced Settings for the Streamline Plot

Under **Advanced** set these general settings. See also Advanced Section Setting Effects.

- The **Integration tolerance** field default is 0.01 for 3D and 0.001 for 2D. Edit to specify how accurately streamlines are computed.
- The **Maximum number of integration steps** field makes sure that the integration does not continue indefinitely. Edit the default (5000) to control when the computation stops.
- The **Maximum integration time** field sets an upper time limit for the integration. The default is infinity (inf).
- The **Stationary point stop tolerance** can be adjusted to make sure the integration stops near a stationary point in the field. The default is 0.01.
- The **Loop tolerance** field default is 0.01. This is a fraction of the mean of the lengths of the bounding box of the geometry. If a streamline gets closer to its start point than this distance, the streamline snaps to its start point and is plotted as a connected loop. See also Method 5: Creating Streamlines with Variable Density and Magnitude Controlled.
- Select the **Allow backward time integration check box** to integrate points from the starting points both in the direction of the vector field and in the opposite direction.
- Select the **Normalize vector field check box** if required.

STREAMLINE POSITIONING SECTION (CONTINUED)

Method 1: Specifying the Number of Streamlines and Start Boundaries

1 Under **Streamline Positioning**, from the **Positioning** list, select **On selected boundaries**.

> The **Selection** section is made available for some data sets when **On selected boundaries** is selected from the **Positioning** list under **Streamline Positioning**.

2 Under **Selection**, select the boundaries from which the streamlines start. By selecting in the **Graphics** window and using the tools in the **Selection** section, select the boundaries for the starting positions for the streamlines.

3 Enter the **Number** of streamlines (the default is 20).

Method 2: Specifying Points by Entering Coordinates

1 Under **Streamline Positioning**, from the **Positioning** list, select **Start point controlled**.

2 Select **Coordinates** from the **Entry method** list.

3 Enter **x** and **y** (2D), **x**, **y**, and **z** (3D) coordinates (SI unit: m). Also use a scalar value to represent a fixed value for some of the coordinates.

Method 3: Selecting the Specified Number of Start Points in the Geometry

1 Under **Streamline Positioning**, from the **Positioning** list, select **Start point controlled**.

2 Select **Number of points** from the **Entry method** list.

3 Enter the number of **Points** (the default is 20).

4 From the **Along line or plane list**, select **None**.

Method 4: Creating Streamlines with Uniform Density

The algorithm saturates the entire domain with evenly spaced streamlines.

1 Under **Streamline Positioning**, from the **Positioning** list, select **Uniform density**.

2 Enter the **Separating distance** between the streamlines (the default is 0.05).

The value for the separating distance is a fraction of the mean of the lengths of the bounding box of the geometry. In this case, a streamline stops whenever it gets too close to another streamline or itself (or if any of the general termination criteria specified in the **Advanced** section is fulfilled).

3 The **Advanced parameters** list defaults to **Automatic**. If required, select **Manual** to edit these parameters—**Boundary element refinement**, **Fraction of streamline length to ignore**, **Starting distance factor**, **Terminating distance factor**, or **First start point**.

- Edit the **Boundary element refinement** if streamlines do not behave as expected near boundaries on a coarse mesh—try increasing this number. It is a measurement of the density of points on the boundaries used to set up the structure and is used to measure distances between streamlines. Refining the mesh in the problematic area can also resolve the problem.
- Edit the value in the **Fraction of streamline length to ignore** field (a fraction 0–1; default value: 0.5) when a streamline is close to itself, typically for spiraling streamlines. This number controls how big part of the streamline, starting from its start point, that the streamline itself is allowed to get close to, and it might in some cases be useful in order to get a less cluttered streamline plot.
- The **Starting distance factor** is a factor multiplied with the distance specified in the **Separating distance** field (as a fraction of the mean of the lengths of the bounding box of the geometry—the default value is 0.05). It sets the minimum distance between streamlines and the start point for the next streamline.

  When the domain is close to be saturated with streamlines, new start points tend to be positioned where the streamline has nowhere to go before it gets too close other streamlines, resulting in short streamlines. The higher the value of this factor, the more it disqualifies the start point and thus reduces the number of short streamlines.
- The **Terminating distance factor** is a factor multiplied with the distance specified in the **Separating distance** field. It sets the minimum distance between any pair of streamlines. Thus, this distance is the minimal distance under which the integration of a streamline stops.
- By default the **First start point** list defaults to **Automatic**, and it sets the start point for the first streamline. It is chosen in the element where the highest value of the velocity of the specified vector field occurs. If required, select **Manual** instead to override the default and enter **x** and **y** coordinates.

Method 5: Creating Streamlines with Variable Density and Magnitude Controlled

To create streamlines with a variable density according to the magnitude of the specified vector field.

1 Under **Streamline Positioning**, from the **Positioning** list, select **Magnitude controlled**.

The **Magnitude controlled** setting gives proper streamline plots only for incompressible flow fields. In this case, the algorithm places the streamlines so that the flow between each pair of adjacent streamlines is the same throughout the domain, giving streamlines that are more dense where the magnitude of the field is high.

2 This step depends if it is a 2D or 3D Component.

For 2D models, enter a **Density** (the default is 20). This value is roughly the number of streamlines. Prior to streamline generation, the software computes a rough estimate of the total flow of the flow field in the model, divides this value with the specified **Density** setting, and uses the resulting value as the flow between each pair of adjacent streamlines.

For 3D models, enter the **Min** (Minimum) **distance** and **Max** (Maximum) **distance** between streamlines (the default **Min distance** is 0.05 and the default **Max distance** is 0.15). These distances are specified as fractions of the mean of the lengths of the bounding box of the geometry. The minimum velocity in the model is mapped to the minimum distance and the maximum velocity to the maximum distance. Thus every point on a streamline and on the boundary has a separating distance associated with it. Given a set of streamlines, the start point for the next streamline is selected using these separating distances.

A streamline stops only if it exits the domain or gets too close to its own start point, using the **Loop tolerance** option in the **Advanced** section (or if any of the general termination criteria specified in the **Advanced** section is fulfilled).

3 If required, from the **Advanced parameters** list, select **Manual** to set advanced parameters as described in Method 4: Creating Streamlines with Uniform Density.

ADVANCED SECTION SETTING EFFECTS

The **Advanced** settings have the following effects:

- When calculating streamlines, the software selects a set of starting points (controlled by the streamline start points and the number of start points).
- The algorithm then finds the vectors of the given vector field at these points by interpolation. It normalizes the vector field if that option is selected.
- The algorithm integrates the points along the direction of the vector using the integration tolerance using a second-order Runge-Kutta algorithm.
- At the new positions, the algorithm finds vector values by interpolation and performs another integration.

This process stops if:

- It reaches a predetermined number of integration steps (controlled by the maximum number of integration steps entry).
- The points end up outside the geometry.
- The points reach a “stationary point” where the vector field is zero. Control the meaning of “zero” with the stationary point stop tolerance.
- It has used a predetermined amount of “time” for integrating (control this parameter with the **Maximum integration time** field).

Finally, the software connects the calculated points for each streamline consecutively with straight lines.

When integrating, the software uses a pseudo-time that has nothing to do with the time in time-dependent problems. Use the massless particle tracing tool to integrate in time-varying fields and to control the real time in stationary fields.

Surface

Use a **Surface** plot to display a quantity on a domain in 2D (🟪) or on a boundary in 3D ( ). Add Deformation, Filter, or Height Expression (2D only) subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plot types.

- Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.
- 2D Plot Group and 3D Plot Group
- Plot Groups and Plots

Table Graph

Add a **Table Graph** (▦) plot to display data from a table with one line per output column. Right-click a **1D Plot Group** or **Polar Plot Group** to add this plot type. First define a table to plot. This plot is also available by selecting **Table Graph** (▦) from the **Table** window's toolbar.

Go to Common Results Node Settings for links to information about these sections: **Legends** and **Coloring and Style**.

DATA

Select a **Table**. In the **x-axis data** list (or θ **angle data** for polar plots), select the column to use as x-axis, select **Row index** to use the table's row indexes (row numbers) as x-axis, or leave it at **Automatic** to let the software determine the input from the data in the table.

The **Plot columns** list controls which columns to plot. **All excluding x-axis** (or **All excluding** θ **angle list** for polar plots) indicates all columns not used in **x-axis data** (or θ **angle data**). Select **Manual** instead to specify which columns to plot in the **Columns** list.

Select a **Transformation** of the data from the table—**None** (the default) to use the data directly without any transformation, or select **Frequency domain** to use FFT to transform the data from a time-dependent solution to the frequency domain.

If **Frequency domain** is selected, the default number of frequencies and frequency range depend on the data, and usually those values do not need changing. To change the values, select the check boxes and enter values for the **Number of frequencies** (the default is 1) and the **Frequency range** in the **Maximum** and **Minimum** fields.

By default, table plots only display the real data in a table, just as other plot types only display real data unless you use the `imag` function in the expression. To display the imaginary part of complex data in a table, when available, select the **Plot imaginary part** check box. This option is only available when a transformation to the frequency domain is not used.

Table Surface

Use a **Table Surface** ( ) plot to visualize the data in a table that represents a matrix of values that are functions of two independent parameters (for example, as a response surface). Right-click a **2D Plot Group** to add this plot type from the **More Plots** submenu. This plot is also available by selecting **Table Surface** ( ) from the **Table** window's toolbar.

> Go to Common Results Node Settings for links to information about these sections: **Title**, **Range**, **Coloring and Style**, and **Inherit Style**.

> When created from the **Table** window toolbar, the 2D plot group with the **Table Surface** plot uses **None** in the **Data set** selection. When you add a Table Surface plot to an existing or new 2D plot group, the data set is typically a solution data set, and the plot group includes a plot of the data set edges. This can make the table surface plot hard to see because it uses parameter values on the x- and y-axis and not the 2D geometry's dimension.

DATA

Select a **Table**. Select an option from the **Plot data** list—**From table** (the default) or **Manual**. If **Manual** is selected, select options from the **x-axis data** and **y-axis data** lists, which contain the parameters that define the rows and columns for the table's matrix data, and from the **Data** list, which corresponds to the **Data** list in the **Table** node for the matrix data. Select the **Plot imaginary part** check box if you want to plot the imaginary part of complex-valued data. For real-valued data, that plot shows a zero imaginary part.

Volume

Use a **Volume** ( ) plot to display a quantity inside a domain in 3D. Add Deformation or Filter subnodes as required. Right-click a **3D Plot Group** to add this plot.

> Go to Common Results Node Settings for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, **Element Filter**, **Shrink Elements**, and **Inherit Style**.

Color Expression

Use a **Color Expression** node ( ) to add coloring (according to an expression that you define) to the shapes or lines defined by a plot. Add this to these plot types—Line Graph, Point Graph, Global, Nyquist Plot, Arrow Volume, Arrow Surface, Arrow Line, Contour, Isosurface, Particle Trajectories, and Streamline.

In the **Model Builder**, add and define a plot group. Right-click the plot node (for example, **Streamline**) and select **Color Expression**.

> Go to Common Results Node Settings for links to information about these sections: **Expression**, **Title**, **Range**, and **Coloring and Style**.

Deformation

Add a **Deformation** node () to deform the plot according to a vector quantity, for example, the displacement field in structural mechanics. You can add a deformation to most 2D and 3D plots—arrow, contour, isosurface, line, slice, streamline, surface, and volume plots. By default, COMSOL scales the deformation to 10% of the geometry. In the **Model Builder**, add and define a **2D Plot Group** or **3D Plot Group**. Right-click the plot node (for example, **Arrow Surface**) and select **Deformation**.

SCALE

Select the **Scale factor** check box to edit the default value for the scale factor.

> Using a scale factor of 1 and equidistant displacements in the x, y, or z direction, you can plot several instances of the geometry side by side to, for example, visualize the solution at some times or for some parameter values. In such a plot you would typically specify the data set in each plot individually and turn off the color legends and titles for each separate plot.

> Go to Common Results Node Settings for links to information about these sections: **Expression** and **Title**.

Filter

Add a **Filter** () subnode to these 2D and 3D plots—arrow, contour, isosurface, line plot, slice, volume, and max/min plots. Filters make it possible to filter (limit) the plot using a logical expression that provides a criterion for which parts of the plot to include. In the **Model Builder**, add and define a **2D Plot Group** or **3D Plot Group** with one or more plot nodes. Right-click the plot node (for example, **2D Surface**) and select **Filter**.

ELEMENT SELECTION

Under **Element Selection**, enter a **Logical expression for inclusion**. Enter any logical expression using predefined variables. For example, `x>0` filters the plot to only include the part of the geometry where $x > 0$.

Select the **Element nodes to fulfill expression**—**All** (the default), **At least one**, or **At least one but not all**. If the default, **All**, is kept, it includes all elements for which all the element nodes (that is, the entire element) fulfill the criterion in the logical expression.

- Select **At least one** to include all elements for which at least one element node fulfills the criterion in the logical expression (that is, elements that fully or partially fulfill the expression).
- Select **At least one but not all** to include all elements for which at least one of the element nodes but not all of them fulfill the criterion in the logical expression. The last option is useful for making a filter that shows the plot for a zone around a boundary where the logical expression becomes true.

Height Expression

The **Height Expression** subnode () introduces 3D height on a 2D surface plot, 2D table surface plot, 2D contour plot, or 2D line plot. Add it to make the height of the plot represent a scalar quantity. 2D Surface, 2D Contour, 2D Line, and 2D Table Surface plots support the height expression attribute. In the **Model Builder**, add and define a 2D **Surface**, **Contour**, **Line**, or **Table Surface** plot then right-click the plot node and select **Height Expression**.

As a subnode to 2D Histogram ( ) plots, the **Height** node settings window does not have an **Expression** section and only **Automatic**, **Manual**, and **None** are available as a **Title type**.

EXPRESSION

Under **Expression**, the **Height data** defaults to **From parent** to use the same data set as the parent plot it belongs to. If **Expression** is selected instead, see Expressions and Predefined Quantities.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

SCALE

For a manual scaling of the height data, select the check box and enter a **Scale factor** to control the height of the added 3D plot. Enter an **Offset** (default value: 0) or use the associated slider to control the base location (relative to the 2D surface's level).

VIEW

The **Height Expression** subnode makes the plot a 3D plot, which needs a 3D view for the grid, camera, lighting, and other 3D view settings. Select the 3D view to use from the **View** list. The default is **Automatic**, which creates a 3D view if needed. Alternatively, select one of the existing **View 3D** nodes in the model.

Derived Values and Tables

About Derived Values

You can integrate or compute the average, maximum, or minimum of any quantity to compute derived quantities such as total flux, charges, inductances, reaction forces, and average, maximum, and minimum values.

Use **Derived Values** (8.85 e-12) to define evaluations of numerical results—globally, in a point, or integrated quantities. For 2D and 3D plots, you can also get numerical results directly in a table by clicking the plot.

For all derived values you can also apply an operator on a data series (from a parametric or Time Dependent study) to compute, for example, the temporal average of a quantity in a point of the domain for which a time-dependent solution is computed. In addition to the average, you can also compute the integral, maximum, minimum, RMS (root mean square), standard deviation, or variance of the data series. The derived values nodes use data sets (typically solution data sets) that provide the data from which the derived values are computed. If needed, specify the frame and geometry to use in the data set's settings window.

GETTING NUMERICAL RESULTS DIRECTLY

For 3D and 2D models, the numerical value of the current plot can be displayed by clicking anywhere in the model geometry.

For a 3D model, the value is for the point where a ray projected from the point clicked hits the geometry.

COMSOL displays the value at that point along with the point's coordinates in a row in an **Evaluation 2D** or **Evaluation 3D** table in the **Table** window. Each click adds a row to the table. From the **Table** window you can plot or copy the table data to a clipboard like any other table.

- See Derived Value Types for links to the data sets.
- Common Results Node Settings

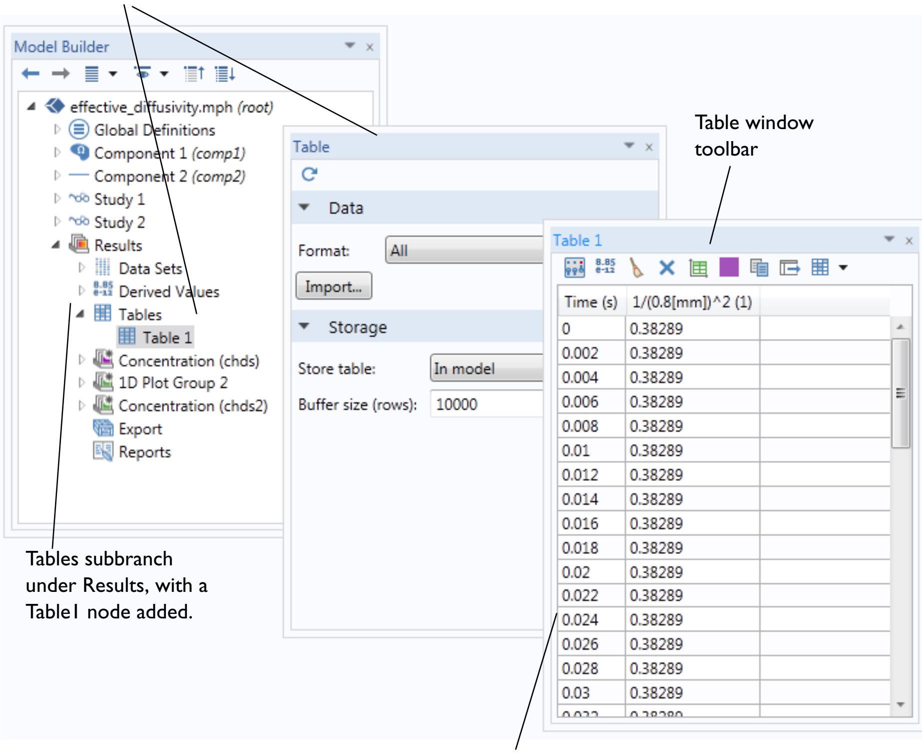
The Table Window and Tables Node

The **Table** window displays the results from integral and variable evaluations defined in Derived Values nodes or by probes and stored in Table nodes.

A **Tables** subbranch (▦) is found under the **Results** main branch and contains all tables defined in the model. Table nodes can also result from evaluating Derived Values. When you select a Table node to display, the window name changes to the name of the Table (**Table 1**, as in Figure 20-2, for example).

- For Windows users, from the **Home** ribbon select **More Windows>Table**.
- For cross platform (Mac and Linux) users, select **Windows>Table**.

Table1 node settings window.



Format: All

Store table: In model

Buffer size (rows): 10000

Time (s)	1/(0.8[mm])^2 (1)
0	0.38289
0.002	0.38289
0.004	0.38289
0.006	0.38289
0.008	0.38289
0.01	0.38289
0.012	0.38289
0.014	0.38289
0.016	0.38289
0.018	0.38289
0.02	0.38289
0.022	0.38289
0.024	0.38289
0.026	0.38289
0.028	0.38289
0.03	0.38289

Table window toolbar

Tables subbranch under Results, with a Table1 node added.

Table window generated from the Results analysis for the Table1 node. The Table window can also be simply for the Tables node displaying results.

Figure 20-2: A example of the Tables subbranch with a Table node, and the associated settings and Table windows.

TABLE WINDOW TOOLBAR AND MENU OPTIONS

When a **Table** (▦) is generated and displays in the **Table** window, a variety of editing options are available as listed in Table 20-9 and shown in Figure 20-2.

The Table window's toolbar control settings are specific to the Table window in general and not to a specific table.

TABLE 20-9: TABLE WINDOW EDITING BUTTONS

BUTTON	NAME	DESCRIPTION
8.85 e-12	Full precision	Click to display as many significant digits as possible.
	Clear table	Click to clear the data from the table, but keep the table itself. Click the Evaluate button (=) to regenerate the table data.
✕	Delete table	Click to delete the table. There is no undo. If required, click the Evaluate button (=) to regenerate the table.

TABLE 20-9: TABLE WINDOW EDITING BUTTONS

BUTTON	NAME	DESCRIPTION
	Plot	Click to plot the table in the **Graphics** window.
	Copy table and headers to clipboard	Click the button or right-click anywhere in the table and select this option from the context menu. You can then paste the table’s data and headers in a spreadsheet, for example.
	Export	Click to export the table to a text file in a spreadsheet format or to a Microsoft Excel Workbook (*.xlsx) if the license includes LiveLink™ *for* Excel®. When saving to a Microsoft Excel Workbook, an Excel Save dialog box opens where you can specify the sheet and range and whether to overwrite existing data and include a header.
	Delete Column	Right-click a column header or anywhere in the table, and select Delete Column. There is no undo. If required, click the Evaluate button (=) to regenerate the table.
	Copy Selection to Clipboard	Select the rows to copy, then right-click anywhere in the table and select this option from the context menu, or press Ctrl+C.
	Copy Selection and Headers to Clipboard	Select the rows to copy, then right-click anywhere in the table and select this option from the context menu.
	Copy Table to Clipboard	Right-click anywhere in the table and select this option from the context menu.
	Copy Table and Headers to Clipboard	Right-click anywhere in the table and select this option from the context menu.

For a surface integration example that includes tables, see Effective Diffusivity in Porous Materials: model library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**.

- Results Analysis and Plots
- Derived Values and Tables

Derived Value Types

In the **Model Builder**, under **Results**, right-click **Derived Values** (8.85 e-12). Select an option from the list and continue defining each derived value (see Table 20-10).

TABLE 20-10: DERIVED VALUE TYPES

ICON	LINK TO SECTION	DESCRIPTION
8.85 e-12	Point Evaluation	To evaluate expressions or variables defined in a point.
8.85 e-12	Global Evaluation	To evaluate the numerical value of a global variable.

TABLE 20-10: DERIVED VALUE TYPES

ICON	LINK TO SECTION	DESCRIPTION
8.85 e-12	Global Matrix Evaluation	To define the evaluation of the numerical values for a global matrix variable such as S-parameters in a model with several ports activated as a parametric sweep and a frequency-domain study.
8.85 e-12	Particle Evaluation	To evaluate an expression for all, or a subset of, the particles in a particle tracing model.
▨	System Matrix	To evaluate an Assemble or Modal node to a table.
	AVERAGE SUBMENU	Volume Average, Surface Average, and Line Average
AV	Volume Average	To evaluate an average over a set of domains in 3D models.
AV	Surface Average	To evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
AV	Line Average	To evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.
	INTEGRATION SUBMENU	Volume Integration, Surface Integration, and Line Integration
∭	Volume Integration	To evaluate an integral over a set of domains in 3D models.
∬	Surface Integration	To evaluate an integral over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
∫	Line Integration	To evaluate an integral over a set of domains in 1D, boundaries in 2D, or edges in 3D.
	MAXIMUM SUBMENU	Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum
MAX	Volume Maximum	To evaluate the maximum over a set of domains in 3D models.
MAX	Surface Maximum	To evaluate the maximum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
MAX	Line Maximum	To evaluate the maximum over a set of domains in 1D, boundaries in 2D, or edges in 3D.
	MINIMUM SUBMENU	Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum
MIN	Volume Minimum	To evaluate the minimum over a set of domains in 3D models.
MIN	Surface Minimum	To evaluate the minimum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
MIN	Line Minimum	To evaluate the minimum over a set of domains in 1D, boundaries in 2D, or edges in 3D.

Volume Average, Surface Average, and Line Average

The derived average values are useful for calculating averaged quantities for each solution in a data set (a time-dependent solution, for example). Also apply an integral, maximum, or other operation to compute the maximum of an averaged quantity, for example.

Under **Results** right-click **Derived Values** (8.85e-12) and from the **Average** submenu select:

- **Volume Average** (AV) to evaluate an average over a set of domains in 3D models. The result of the evaluation is stored in a **Table** node and displayed in the **Table** window.
- **Surface Average** (AV) to evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Average** (AV) to evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.

> Go to Common Results Node Settings for information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operations**.

> For a line average example, see Tubular Reactor: model library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

Volume Integration, Surface Integration, and Line Integration

The derived integration values are useful for calculating integrated quantities for each solution in a data set (a time-dependent solution, for example). Also apply an average, maximum, or other operation to compute the average of an integrated quantity, for example. Under **Results** right-click **Derived Values** (8.85e-12) and from the **Integration** submenu select:

- **Volume Integration** (∭) to evaluate an integral over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Table** window.
- **Surface Integration** (∬) to evaluate an integral over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Integration** (∫) to evaluate an integral over a set of domains in 1D, boundaries in 2D, or edges in 3D. Integrate over any data set of the right dimension. For example, make a volume integration of a 2D revolved data set or a surface integration of a cut plane.

> Go to Common Results Node Settings for information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operation**.

> For volume integration examples, and if you have the:
>
> - AC/DC Module, for both a volume and surface integration example, see Eddy Currents: model library path **ACDC_Module/Inductive_Devices_and_Coils/eddy_currents**.
> - RF Module, see Microwave Oven: model library path **RF_Module/Microwave_Heating/microwave_oven**.

> For a surface integration example see Effective Diffusivity in Porous Materials: model library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**.

For line integration examples, and if you have the:

- AC/DC Module, see An RFID System: model library path **ACDC_Module/Inductive_Devices_and_Coils/rfid**.
- CFD Module, see Filling of a Capillary Channel—Level Set: model library path **CFD_Module/Multiphase_Tutorials/capillary_filling_ls**.
- Microfluidics Module, see Filling of a Capillary Channel—Level Set: model library path **Microfluidics_Module/Two-Phase_Flow/capillary_filling_ls**.
- Heat Transfer Module, see Cavity Radiation: model library path **Heat_Transfer_Module/Verification_Models/cavity_radiation**.

Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum

The derived maximum and minimum values are useful for calculating maximum or minimum quantities for each solution in a data set (a time-dependent solution, for example). Also apply an average, maximum (or minimum), or other operation to compute the maximum or minimum over the entire data set for a quantity, for example. Under **Results** right-click **Derived Values** (8.85 e-12) and from the **Maximum** or **Minimum** submenus select:

- **Volume Maximum** (MAX) or **Volume Minimum** (MIN) to evaluate a maximum or minimum value over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Table** window.
- **Surface Maximum** (MAX) or **Surface Minimum** (MIN) to evaluate a maximum or minimum value over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Maximum** (MAX) or **Line Minimum** (MIN) to evaluate a maximum or minimum value over a set of domains in 1D, boundaries in 2D, or edges in 3D.

ADVANCED

Choose to find the maximum or minimum of the real part or the absolute value, which are different for complex-valued data. Choose **Real part** (the default) or **Absolute value** from the **Find maximum of** or **Find minimum of** list.

Select an **Element refinement** (default: 5; the element refinement is the number of partitions of an element edge) to adjust the accuracy of the minimum or maximum values.

Go to Common Results Node Settings for information about these sections: **Data**, **Selection**, **Expression**, and **Data Series Operation**.

For a volume maximum example, and if you have the Nonlinear Structural Materials Module, see Polynomial Hyperelastic Model: model library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/polynomial_hyperelastic**.

Point Evaluation

Use **Point Evaluation** (8.85 e-12) to define the evaluation of a variable or an expression in a point. The result is stored in a **Table** node and displayed in the **Table** window.

- Go to Common Results Node Settings for information about these sections: **Data**, **Expression**, and **Data Series Operation**.
- Table

Global Evaluation

Use a **Global Evaluation** (8.85 e-12) to define the evaluation of the numerical value of a global variable.

- Go to Common Results Node Settings for information about these sections: **Data**, **Expression**, and **Data Series Operation**.
- Table

Global Matrix Evaluation

Use a **Global Matrix Evaluation** (8.85 e-12) to define the evaluation of the numerical values for a global matrix variable such as S-parameters in a model with several ports activated as a parametric sweep and a frequency-domain study. The **Table** window then displays all values for all frequencies in a parametric sweep.

- Go to Common Results Node Settings for information about these sections: **Data** and **Expression**.
- Table

DATA SERIES OPERATION

Select an option for operations on the data series for the inner solutions—typically a frequency sweep or time series from a study—and the outer solutions—the parametric sweep (for ports in electromagnetics, for example)—from the **Inner solutions** and **Outer solutions** lists:

- Select **None** to use the computed value as it is (the default).
- Select **Average** to use the average of the computed values for the inner solutions or the outer solutions.
- Select **Sum** to use the sum of the computed values for the inner solutions or the outer solutions.

The **Ignore NaN** check box is selected by default so that the results tables do not include NaNs (Not-A-Number values). If you want to see values that a NaNs, clear this check box.

TRANSFORMATION

Apply a transformation operation to compute the inverse of the matrix variable or to convert between the impedance matrix, **Z**, the admittance matrix, **Y**, and the S-parameter matrix S (available for the AC/DC Module and RF Module). From the **Transformation** list, choose one of the following transformations:

- **None** (the default).
- **Inverse**.
- **From S to Y**. For this transformation, also specify the **Characteristic admittance** (SI unit: S). The default value is 1 S.

- **From S to Z**. For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is 1 Ω.
- **From Y to S**. For this transformation, also specify the **Characteristic admittance** (SI unit: S). The default value is 1 S.
- **From Y to Z**.
- **From Z to S**. For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is 1 Ω.
- **From Z to Y**.

The transformation operations are only applicable for square matrices.

System Matrix

Use a **System Matrix** ( ) derived values node to evaluate a matrix defined in an **Assemble** or **Modal Solver** node to a table.

SOLUTION

Select a **Solution** from the list to specify which **Solver** branch to tabulate the system matrix for. From the **Solver feature** list, select the **Assemble** or **Modal Solver** node that computes the system matrix.

OUTPUT

Select a **Matrix**—**Stiffness matrix** (the default), **Damping matrix**, **Mass matrix**, or any other system matrix that the selected **Assemble** or **Modal Solver** node computes.

First select the system matrices to assemble or compute in the settings window for the **Assemble** or **Modal Solver** node and then compute the solution so that the solver stores these system matrices. Selecting a system matrix that has not been computed and stored results in an error.

Select a **Format**—**Sparse** (the default) or **Filled**. The system matrices can become very large but are usually sparse (most matrix elements are zero). If **Filled** is selected, the preference setting for the maximum filled matrix size (the default is 100) prevents the creation of a table with a very large matrix. Typically, when this is an Assemble node, a filled matrix output can only be used for very small models or when using reduced matrices.

When the System Matrix is a Modal Solver, then the matrices of interest are typically small and filled.

- Introduction to Solvers and Studies and Solution Utility Nodes
- Derived Values and Tables

Particle Evaluation

Use a **Particle Evaluation** ( ) derived values node to define the evaluation of the numerical value of quantities computed by one of the particle tracing interfaces. This node requires the Particle Tracing Module.

- Go to Common Results Node Settings for information about these sections: **Data** and **Expression**.
- Derived Values and Tables

DATA

Select a **Data set**. Only the **Particle** data sets ( ) are available for selection. If no particle data set is available then the only option is **None**.

The **Select via** option is used to select the times at which the particle evaluation occurs. When **Stored output times** is selected, the **Time** list is populated with the output times from the solver, which is linked to the particle data set. When **Interpolation times** is selected, then enter an array of times for which the particle evaluation should be performed at.

EVALUATION

To specify the particles to evaluate the expression for, the **Particles to evaluate** list contains the following options:

- Select **All** (the default) to evaluate the expression for all particles in the simulation.
- Select **Fraction** to evaluate for a fraction of all particles. Enter a scalar value between 0 and 1 in the **Fraction of particles** field. The default value is 1, which means that the evaluation includes all particles.
- Select **Number** to specify a number of particles to evaluate for in the **Number of particles** field. The default value is 100 particles. If the particle simulation contains fewer particles than the specified number, all particles are included.

The Results Table

Click the **Evaluate** button (=) or right-click the **Derived Values** node and select **Evaluate All** (=) or **Clear and Evaluate All** ( ).

The first column in the table is a list of the **Time** values selected from the **Data** section. There are *N* additional columns, where *N* is the number of particles chosen in the **Evaluation** section.

The values in each column correspond to the supplied **Expression** for all the selected **Time** values. Each column contains *M* rows, where *M* is the number of **Time** values selected in the **Data** section. By default four digits are displayed; click the **Full precision** button ( ) to display as many significant digits as possible. If required, the precision level can be changed from the The Preferences Dialog Box under **General**.

Table

Tables can store the results of **Derived Values** and results from probes, for example. The results are displayed in the **Tables** window, which by default is located below the **Graphics** window. To add a **Table** node, if not already created, right-click the **Tables** ( ) node and select **Table**. You can also add a table by, in the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, clicking the **Evaluate (New Table)** button (=). This evaluates the node and stores the result in a new table. This button is also used as the **Evaluate** button.

A **Table** node is added under **Tables**. Click the node to display a table with the selected integration node's description and values in the **Table** window.

There are three ways to evaluate a **Derived Values** node and put the result in a **Table**:

- In the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, click the **Evaluate** button (=).
- In the **Model Builder**, right-click the specific **Derived Values** node (for example, **Volume**, **Surface**, or **Line Integration**) and select **Evaluate**.
- In the **Model Builder**, right-click the **Derived Values** node, and select **Evaluate All** (=) to evaluate all the **Derived Values** nodes. This appends the results to any existing tables. Select **Clear and Evaluate All** (=) to first clear the affected tables and then evaluate all the **Derived Values** nodes.

In addition, you can specify the output table for data from a probe in the **Table and Windows Settings** section of the probe settings window. Also, for nested parametric sweeps you can add an accumulated probe table, which provides a matrix of values where the rows and columns represent two independent parameters.

To delete all tables, right-click the main **Tables** ▦ node and select **Delete All** (✕). To clear the contents of all tables, right-click the main **Tables** ▦ node and select **Clear All**.

DATA

If applicable (for accumulated probe tables, for example) select a data format from the **Format** list—**All** (the default) or **Filled**. **All** displays all data in the table. For many tables this is the only available format.

Select **Filled** to create a matrix of data from, for example, a nested parametric sweep with independent parameters. Filled tables can only be produced by studies that have **All combinations** selected in their parametric sweeps. The filled tables make it possible to retrieve data for a pair of parameters on a matrix format and can be used to make response surface plots. You can get filled tables from some probes or from using derived values when all solutions are selected. Basically, you get filled tables when the input parameters in a parametric sweep (such as time) represent a full outer product. In some cases, such as eigenvalue solutions or if the solver is interrupted, the data is not filled. Also, if you modify the table by, for example, evaluating more than once into the same table or delete columns, the table data is no longer filled.

- Select the parameters to use for the rows and the columns from the **Rows** and **Columns** lists, and select the data (probed quantity, for example) to use in the table from the **Data** list.
- Under **Column headers**, select a **Column description** (in addition to the parameter values): **None**, **From data** (the default), or **Manual**. If you select **Manual**, type the description in the **Description** field.

Click the **Import** button to import data from, for example, a spreadsheet. In the **Import** dialog box that opens, select a data file to import.

STORAGE

If desired you can store the table data on file instead of or in addition to storing it in the model. Select a way to store the table data from the **Store table** list:

- Select **In model** (the default) to store the table data in the model.
- Select **On file** to store the table data in the model to a file that you select by first clicking **Browse** or enter into the **Filename** field. This makes it possible to, for example, keep track of more data using probes while solving without having to put all the data in the model.
- Select **In model and on file** to store the table data both in the model and on file.

For the storage in the model, you can specify the buffer size in the **Buffer size (rows)** field, The default is 10,000 rows.

The imported data replaces any existing data in the table.

Exporting Data and Images

Export Types

After a model is completed, you can add various components to the **Export** branch and then generate outputs (animations, data, images, player, or export), or export the information to your computer as image, movie, or data files for use in external documents or for other purposes. In the **Model Builder**, under the **Export** node, right-click and select an option as listed in Table 20-11.

TABLE 20-11: EXPORT TYPES

ICON	LINK TO SECTION	DESCRIPTION
🎞	Animation	To define and export a movie or series of images based on a plot group. Play the animation in a web browser or use it in presentations or on a web site.
▤	Data	Exports numerical data to file. Data export operates directly on data sets. It is also possible to export mesh data.
△	Mesh (Export)	To export a mesh defined by a data set to file.
▦	Table	To export the contents of a table to file.
🖼	1D Image, 2D Image, or 3D Image	To export plot images from a 1D Plot Group, 2D Plot Group, and 3D Plot Group.
▶	Player	To create interactive animations.
🟥	Plot	To export a plot from a 1D Plot Group, 2D Plot Group, or 3D Plot Group.

About the Sectionwise Data Format for Data Export

When exporting data on the sectionwise data format, the program evaluates the entered expressions at a number of points in each mesh element. For example, evaluating in Lagrange points of order 1 means that the expressions are evaluated at the vertices of each mesh element. When a vertex is shared by more than one mesh element (as is typically the case) this means that the expressions are evaluated several times at that coordinate, but using the shape functions in the different mesh elements. The values of these evaluations at the same point might not be equal, depending on the expression being evaluated. In particular, derivatives are typically discontinuous across mesh element boundaries and usually have different values.

Once all the evaluations have been made, the data are checked for duplicate values (that is, evaluations with the same coordinates and the same values of the expressions). Such duplicates are removed before the data is exported to file. With smoothing turned on, a smoothed variant of the derivative is evaluated, which is continuous across mesh element boundaries, so in such cases there are many duplicates. When evaluating at Gauss points, the evaluation points are always in the interior of mesh elements, so there are never any duplicates.

To avoid the removal of duplicates, you can export several expressions to the same file, and then the values of all expressions must agree to be considered duplicates. Another way to ensure that no duplicates are removed is to add the variable `meshelement` to the list of expressions.

- Data
- See Table 20-11 for links.

Animation

Use **Animation** (▦) to define and export a movie or series of images based on a plot group. Play the animation in a web browser or use it in presentations or on a website. Use this node, for example, to export multiple images for different time steps or eigenvalues.

! Use Windows Media Player to play AVI files.

SCENE

Select a **Subject**, which is one of the plot groups previously defined, or **None**.

OUTPUT

Select an **Output type**—**Movie** (the default) to generate a single movie file containing all the images; or **Image sequence** to generate multiple image files, one for each frame.

- If **Image sequence** is selected:

  Enter a **Filename** including a path to save it to your computer, or click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the desktop and enter a **Filename** in the **Export Image Sequence** dialog box then select a image file type from the **Save as type** list—**.png**, **.bmp**, or **.jpg**.

  The text entered in the **Filename** field is used for all the images generated. For example, if `image` is entered, select **.png** as the file type, and if there are 11 frames in the movie, 11 files are created: `image01.png`, `image02.png`, …, `image11.png`.
- If **Movie** is selected, select a **Format** for the movie: **GIF** (the default), **Flash**, or **AVI**. For any movie format, enter a path and include a **Filename**. Or click **Browse** and navigate to where you want to **Save** the output. Also enter a number of **Frames per second** (the default is 10).

Enter information into these fields based on the **Format** for the movie:

- If **Flash** is selected, the **Interpolate between frames** check box is selected by default. Click to clear the check box if you do not want interpolation between the frames.
- If **AVI** is selected, enter a **Quality**, a scalar value between 0 and 1. The default is 0.75.
- For **Flash** and **GIF**, select the **Open in browser** check box to launch the default web browser to view the output Flash or GIF file.

! AVI is a file format that can contain video encoded in different ways. However, the AVI format is not supported on Windows XP. For movies using this format, you might also need to change the codec used for animations (in the **Graphics** section of the **Preferences** dialog box).

ANIMATION EDITING

Control how the software creates the frames for the animation sequence. Select a **Sequence type**— Stored Solutions (the default), Result Parameter, or Dynamic Data Extension. Define the **Sequence type** parameters as required.

> Each selected animation sequence component creates a frame in the movie or an individual image file.

Stored Solutions

The default, **Stored solutions**, is useful to animate time-dependent solutions or across the eigenmodes for an eigenvalue/eigenfrequency solution or across the parametric solutions for a solution from a parametric sweep. If **Stored solutions** is selected:

From the **Loop over** list, select the steps or parameter values to **Loop over**— **All solutions** (the default), or if applicable, a parameter or combination of parameters, or any parameter in a **Parametric Sweep** study, or the **Time**. If **All solutions** is chosen, a list appears that contains all combinations of parameter values and times (if applicable).

- **Parametric Sweep Study**: For parametric sweep studies where there are multiple inner solutions (for example, a parameter sweep around a time-dependent solution), **Inner solutions** and **Outer solutions** are also available, typically corresponding to **Time** and parameter values, respectively. Typically, an animation shows variations looping over a parameter, frequency, or time; animating all solutions might be useful to get an overview of all solutions that the model contains.
  - If you choose to loop over **Inner solutions**, select the **Parameter value** set to animate, then the time steps, which you can select, using the **Select via** list, as **Stored output times** (a **Time** list of all stored times), or **Interpolated times** (a text field where you can specify any times within the time range directly. See Volume Integration, Surface Integration, and Line Integration and Global for more information about the inner and outer solutions.
  - If you choose to loop over **Outer solutions** (that is, the parameters from the parametric sweep), the **Parameter values** list contains all combinations of parameter values. Select as required, then select an option from the **Inner type** list—**First**, **Last** (the default), or **All**.
- If a **parameter or combination of parameters** are selected to loop over, choose an option from the **Parameter selection** list: **All** (the default), **From list** to select from a list of all parameter combinations, or **Manual**, to enter a range of parameter value indices directly (or click the **Range** button ( )).

  If the model includes other parametric sweeps or frequency sweeps, specify the value of those parameters for the frames in the animation in separate **Parameter value** lists. Also, if the model includes a time-dependent solution, select a time step from the **Time** list, or select **Interpolated** to specify any time within the time span in the text field that appears.
- If you select to loop over the **Time**, for time-dependent problems, choose an option from the **Time selection** list— **All** to use all time steps, **From list** to select from a list of all time steps, **Manual** to enter a range of times as indices directly, or select **Interpolated** to enter **Times**. If the model contains a parametric sweep, select an option from the **Parameter value** list.

Result Parameter

Use a **Result parameter** to animate the changes resulting from a sweep of the values for a defined global parameter (found under **Global Definitions>Parameters**). Using a parameter you can, for example, animate a sweep of the position of a slice across the geometry in a slice plot.

If **Result parameter** is selected, choose a **Parameter** from the list (or select **None**), which contains all global parameters, and define an interval for the parameter values using the **Start** and **Stop** fields.

Dynamic Data Extension

Use a **Dynamic data extension**, for example, to animate the dynamics of an eigenmode in an eigenfrequency or eigenvalue solution. In such a dynamic data extension, the full harmonic cycle (the default) is the normal choice. You can also use it to animate a stationary solution even if there is no obvious interpretation of the animation.

If **Dynamic data extension** is selected, and when animating static and eigenvalue solutions, select a **Cycle type**:

- **Full harmonic**—a full sine wave (the solution phase grows linearly from 0 to 360°)
- **Half harmonic**—half a sine wave (the solution phase grows linearly from 0 to 180°)
- **Linear**—a linear ramp ($\mathrm{Re}(e^{i\alpha}$), where α is the phase, grows linearly from 0 to 1)

  The cycle starts from the angle specified in the **Solution at angle (phase)** field when defining a **Solution** data set.

FRAMES

- If the selected **Sequence type** is Stored Solutions, choose a **Frame selection**— **All** (to play all solutions in the stored solution) or **Number of frames**. For any sequence type, or if number of frames is selected here, enter the **Number of frames**. The default is 25 frames.
- From the **Size** list, select **Manual** (the default) or **Current**, which uses the current size of the **Graphics** window.
- If **Manual** is chosen, select the **Lock aspect ratio** check box to keep the original animation width and height. In the **Width** and **Height** fields, enter the number of pixels (px) for the generated image size. The default value is 640 pixels (width) by 480 pixels (height).
- If required, select the **Record in reverse order** check box.

LAYOUT

- By default, the **Title**, **Legend** (1D graphs) or **Color legend** (2D plots), **Axes**, and **Logotype** (1D and 2D plots) or **Title**, **Color legend**, **Grid**, **Axis orientation**, and **Logotype** (3D plots) parts of the graphics are included. To edit the default, select the **Include** check box and click to clear or select one or several of the available check boxes.
- Enter a **Font size** (pt) for the text in the animation frames. The default is 9 pt.
- Select a **Background**—**Color** (the default) or **Current**. If **Color** is selected, click **Color** to select a **Custom color** background to replace the default, which is white. Select **Current** to use the background in the plot group, which is a blue gradient background for 3D plots and white for 2D and 1D plots.

ADVANCED

If needed, adjust settings for the resolution, antialiasing, and synchronization of the scales in the animation frames.

- Enter a **Resolution** for the images in the animation. The default is 96 DPI.
- Select the **Antialiasing** check box to reduce stairstep-like lines and to smooth lines and edges.
- By default, the **Synchronize scales between frames** check box is selected, which means that all frames in the animation use the same color scale, isosurface levels, deformation scale, and so on. This synchronization makes areas with the same solution values keep the same color, for example, during the entire animation.

  Click to clear the check box to make the scales and levels adapt to the solution in each frame. This can be useful, for example, for time-dependent simulations of transient phenomena where the magnitude of the solution changes significantly during the time stepping. With the synchronization active it can then be difficult to distinguish small variations in the solution.

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**. The animation file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Parametric Sweep and Introduction to Solvers and Studies

Data

Use **Data** ( ) to export numerical data to a file. Data export operates directly on data sets. You can use different types of evaluation points other than the ones in the data set (for example, a grid) and export the data in spreadsheet or sectionwise formats.

To export data, you can right-click **Export** ( ) and select **Data** ( ) or right-click any data set node, for example, **Solution**, and select **Add Data to Export**. Click the **Data** node under **Export**.

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

- Go to Common Results Node Settings for information about these sections: **Data** and **Expressions**.
- For detailed information about Data Formats, see the *COMSOL API Reference Manual*.

OUTPUT

Enter a **Filename** including a path to save the data file to your computer or click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the desktop and enter a **File name** in the **Export Data** window. It is saved as a .txt file.

Select the **Points to evaluate in**—**Take from data set** (the default), **From file**, **Grid**, or **Regular grid**. Depending on the selected type of points to evaluate in, various settings are available.

Take from Data Set

The default **Take from data set** uses the data points for the data in the data set. Select a **Data format**—**Spreadsheet** (the default) or **Sectionwise**. Spreadsheet data is useful to use the data in spreadsheet applications and sectionwise data format is useful for unstructured interpolation because it contains the exact mesh used to perform the interpolation.

For either choice, select a **Space dimension**—**Take from data set** (the default), **Global**, **0**, **1**, **2**, or **3**. Then select a **Geometry level**—**Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point** (availability is based on the model space dimension).

If **Spreadsheet** is selected (and if required), choose the **Transpose** check box to transpose the data from columns to rows.

From File

If **From file** is selected, it uses coordinates from a data file. Then enter a **Coordinate filename** for a text file with the coordinates for the data output, or click **Browse** to locate the file.

Grid or Regular Grid

If **Grid** or **Regular grid** is selected, it uses a grid to define the points to evaluate in. Select a **Data format**— **Spreadsheet** (the default) or **Grid**. Spreadsheet data is useful for using the data in spreadsheet applications, whereas the grid data format is more compact and can be useful to store data that can be imported into another model.

If **Grid** is selected as the **Data format**, also specify the **x**, **y**, and **z** coordinates for the grid points in the fields, or, for the **Regular grid**, specify the **Number of x point**s, **Number of y points**, and **Number of z points** for the regular grid in the fields (default: 10 points in each direction).

ADVANCED

- The **Include header** and **Full precision** check boxes are selected by default. Click to clear the check box if you do not want to include a header, or to limit the precision in the output to six significant figures (and which provides an output that contains all significant figures for data stored as double-precision numbers).
- By default the data is unsorted. Select the **Sort** check box to sort the data by increasing x, y, and z coordinates.
- The **Evaluate in** list is only available for data from Solution data sets. From the **Evaluate in** list select **Lagrange points** (the default) or **Gauss points** to specify where COMSOL evaluates the data—the nodes of the Lagrange elements or in the Gauss points for the Gaussian quadrature, respectively.
- Select a **Resolution**—**Normal** (default), **Finer**, **Fine**, or **Custom**. If **Custom** is selected, enter a **Lagrange-element node-point order** (the default is 1). Use a higher node-point order for a finer resolution.
- Select a data **Smoothing** method—**None**, **Internal** (the default, for smoothing within domains but not across interior boundaries), or **Everywhere**.

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Expressions and Predefined Quantities

Mesh (Export)

Use the **Mesh** ( ) node to export a mesh to file. Mesh export operates directly on data sets and exports the mesh in the frame specified by the data set. The mesh data can also be exported from data sets that contain meshes, for example, a **Mesh** data set or a **Solution** data set. Right-click the **Export** node and select **Mesh**. In the **Mesh** node’s settings window, follow these steps to export mesh data:

DATA

Select a **Data set**. The **Data set** list contains the solution and mesh data sets previously defined. Select **None** to not export any mesh data.

OUTPUT

Select a **File type** from the list of available file formats. Enter a **Filename** including a path to save it to your computer or click **Browse** and navigate to where you want to **Save** the output. You can export the mesh to a COMSOL Native file (binary `.mphbin` or text `.mphtxt`) or to an STL file (`.stl` or text `.stl`) (if a 3D data set is selected).

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**.

Importing and Exporting Meshes

Table

Use the **Table** ( ) node to export the contents of a table to file. A table export stores the data from any of the tables in the model as a text file. To export a table, right-click a **Table** node under **Tables** ( ) and select **Add Table to Export**, or right-click the **Export** node and select **Mesh**.

TABLE

Select a **Table**. The **Table** list contains the all tables in the model. By default, the selection is the one from which you have selected **Add Table to Export** or the first available table if you have added the **Table** node directly under **Export**. Select **None** to not export any table data.

OUTPUT

Enter a **Filename** including a path to save the table data as a file to your computer or click **Browse** and navigate to where you want to **Save** the output and specify the file type as a text file (*.txt), CSV file (*.csv), data file (*.dat), or Microsoft Excel workbook (*.xlsx) from the **Save as type** list. When you save the table data as a Microsoft Excel workbook, you can also specify a **Sheet** and **Range** (by default, those text fields are empty; the program then saves all data), and by default the **Overwrite** check box is selected.

ADVANCED

By default, the **Full precision** check box is selected and exports the table data using full precision for double precision floating-point data. Click to clear the check box to export the data using the table display precision instead.

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **Output** section.

The Table Window and Tables Node

1D Image, 2D Image, or 3D Image

Use the **1D Image**, **2D Image**, and **3D Image** nodes to export plot images as PNG, BMP, EPS, TIFF, GIF, or JPEG image files.

To export a plot right-click **Export** ( ) and select **1D image**, **2D Image**, or **3D Image** or right-click any **Plot Group** node, for example, **3D Plot Group** or **1D Plot Group,** and select **Add Image to Export**. Click the **Image** node under **Export**.

Use the Animation node, for example, to export multiple images for different time steps or eigenvalues.

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

SCENE

Select a **Subject**. The list contains the 1D, 2D, or 3D plot groups previously defined. **1D Image** uses data from 1D Plot Groups, **2D Image** uses data from 2D Plot Groups, and **3D Image** uses data from 3D Plot Groups.

Under **Scene**, for **2D Image** and **3D Image** models, select a **View**—**From plot group** (the default) to use the view from the plot group settings, or select a another view from the list (if available and previously defined).

On the settings window toolbar, click the **Refresh** button ( ) to refresh the **Graphics** window to get a preview of the image to export. This is useful when the **Subject** or **View** selection is changed.

IMAGE

Choose a **Size**—**Manual** to specify the image size manually or **Current** to use the current size of the **Graphics** windows. For either **Size** option, the **Antialiasing** check box is selected by default to reduce stairstep-like lines and to smooth lines and edges. The rest of these settings are available if **Manual** is selected.

- Select a **Unit** of dimension—**Pixels (px)** (the default), **Millimeters (mm)**, or **Inches (in)**.
- Select the **Lock aspect ratio** check box to keep the original image width and height.
- In the **Width** and **Height** fields, enter the number of pixels, millimeters, or inches for the final image size.
- Enter a **Resolution**. The default is 96 DPI (dots per inch).

FILE

Choose an image file **Format**—PNG (the default), BMP, JPEG, TIFF, GIF, or EPS (1D only). If you choose the JPEG format you can also control the quality of the image using a quality measure (scalar number) between 1and 100 (a higher number represents a higher quality). The default value is 92. Select the check box next to **Quality** to enter another quality number.

> JPEG is a format that uses “lossy compression,” so using a low quality measure can make the exported image differ from the original image.

Enter a **Filename** including a path to save it to your computer or click **Browse** and navigate to where you want to **Save** the output. The text entered in the **File name** field is used for all the images generated. For example, if `image` is entered, select **.png** as the file type, and then there are 11 frames in the movie and 11 files are created: `image01.png`, `image02.png`, …, `image11.png`.

LAYOUT

> When importing an image with a transparent background to another Windows application, first save the image as a file rather than saving it to the clipboard. In some cases the transparent background is not preserved if you copy an image via the clipboard.

One additional **Background** option is available for the Image export when using a PNG file format—**Transparent**. Otherwise, see Animation for the rest of the settings.

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **File** section.

Player

A **Player** ( ) creates interactive animations directly in the COMSOL Desktop. To export a player, in the **Model Builder**, right-click **Export** ( ) and select **Player**.

SCENE

Select a **Subject** for the player; the list has the plot groups previously defined, or choose **None**.

ANIMATION EDITING AND ADVANCED

See Animation for the settings.

FRAMES

- If the selected **Sequence type** is **Stored solution**, select **All** or **Number of frames** from the **Frame selection** list. Select **All** to play all solutions in the stored solution.
- Enter the **Number of frames** (if applicable). The default is 25 frames.
- To preview individual frames, enter the **Frame number** or select it using the slider. Observe the geometry in the **Graphics** window to see the **Shown frame** number.

PLAYING

Use this section to adjust some settings that affect the playing of the recorded plots. In the **Display each frame for** field, enter the time to display each frame (in seconds) to control how fast the player runs (default value: 0.1 s). Select the **Repeat** check box to replay the sequence of plots repeatedly instead of playing it just once.

- In the settings window, click **Generate Frame** ( ) (or right-click the **Player** node) to create a series of frames (if more than one **Frame number** is selected).
- Right-click the **Player** node and select **Play** (▶). Watch each **Frame number** cycle from beginning to end in the **Graphics** window. Use the buttons on the **Graphics** window to **Play** (▶), **Stop** (■), and **Next** (⏭) and **Previous** (⏮) to cycle through the animation.

Plot

Right-click **Export** ( ) and select **Plot** ( ) to export a plot from a plot group. Or right-click any plot, for example, the Slice plot in a 3D Plot Group, and then select **Add Plot Data to Export**.

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

PLOT

Select a **Plot group** from the list, which contains any previously defined plot groups. Select a **Plot** to export its data. **Plot Groups** can contain one or more individual plots.

OUTPUT

Enter a **Filename** including a path to save it to your computer or click **Browse** and navigate to where you want to **Save** the output.

From the **Data format** list, select **Spreadsheet** (the default), **Sectionwise**, **STL Binary File (*.stl)**, or **STL Text File (*.stl)** (last two options are only available for Volume, Surface, Slice, Multislice, Isosurface, and Far Field plots). For Streamline plots and Particle Tracing plots you can also control the amount of data to export. By default, the data contains full information about all points for all particles or streamlines.The filename extension is automatically adjusted according to the data format.

Always check the filename extension after selecting the data format.

Select the **Only export start and end points** check box to only include one row with the start and end points for each particle or streamline.

This check box only has an effect on particles plotted as lines because plots of particles as points do not contain full information about the particle trajectories.

ADVANCED

- The **Include header** and **Full precision** check boxes are selected by default. Click to clear the check box if you do not want to include a header, or to limit the precision in the output to six significant figures (and which provides an output that contains all significant figures for data stored as double-precision numbers).
- By default the data is unsorted. Select the **Sort** check box to sort the data by increasing x, y, and z coordinates.

The advanced settings are ignored if you export a plot to an STL file.

Click the **Export** button ( ) in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **File** section.

Reports

About the Report Generator

The Report Generator is a tool for reporting and documenting models created in COMSOL. It creates a record of the entire model including all the settings made during the modeling process. The report is an overview of the model and includes model properties, geometry, physics, mesh, studies, and results and visualization.

Several reports can be created for each model, and you can configure each report by adding, moving, deleting, and disabling the nodes that define the report. The reports are stored with the model, so you can keep generating reports using the previously configured report contents and update the reports when the model changes.

These reports are easy to publish as electronic documents suitable for the Internet or as Microsoft® Word documents (.docx files). The utility generates the HTML report in a file format that makes it possible to customize the report in any HTML editor. A custom style sheet can be used to format the report. You can use HTML tags to create hyperlinks and format the report output in text boxes, headings, and captions.

Generating a Model Report

For a model for which you want to create a report, the following steps describe the general procedure:

1 Right-click **Reports** ( ) and choose a template that creates report nodes that describe the model with a suitable level of detail: **Brief Report**, **Intermediate Report**, or **Complete Report**. You can also choose **Custom Report** and **Document**.

2 In the main **Report** settings window, specify the output format (**HTML** or **Microsoft Word**) and the location for the output file and the associated folder with images and style sheet information. You can also specify the style sheet to use for HTML reports and how to enumerate the sections in the report.

3 The top node is typically the **Title Page** node ( ). In its settings window you can define the title (defaults to the model's file name), an image to use at the top of the report, author and company information, and add a summary and acknowledgments if applicable.

4 Review the structure and contents of the report. You can add, move, edit, disable, and delete structural elements and report contents.

Report Types

For any model you can add one or more reports to the **Reports** branch and then generate model reports for documentation and information about a model. You can create reports using predefined templates that define different levels of detail:

- **Brief Report**: contains an overview of the model with all results and plots but no details about the physics and variables.
- **Intermediate Report**: contains comprehensive information about the model, including the physics settings and variables but not complete information about the underlying equations, for example.

- **Complete Report**: contains all information about the model, including physics interface details such as weak equation expressions and shape functions. This report is suitable for troubleshooting, for example.
- **Custom Report**: contains an initially empty report, which you can configure using the availably report components.

> For all report types, the templates provide a starting point. It is possible to customize all reports by modifying, moving, adding, disabling, and deleting nodes in the reports. You can also switch the level of detail for an existing report, which affects the report nodes added afterward.

The Report Node

The main **Report** node ( ) contains information about the formatting and defaults for the report. Click the **Preview Selected** ( ) or **Preview All** ( ) button to show a preview of the report in the **Preview** window. Click the **Write** button ( ) in the toolbar for the **Report** settings window to create a report. The **Write** option is also available by right-clicking any node in the report. Selecting **Write** from any report node's context menu generates the entire report.

LEVEL OF DETAIL FOR NEW NODES

This section, which is collapsed by default, contains the setting for the level of detail in the new nodes that are added to the report. It is independent of the level of the template used to create the initial report contents. From the **Use default settings for** list, select **Brief**, **Intermediate** (the default), or **Complete** to specify a level of detail for new report nodes that is the same as the corresponding report templates use.

FORMAT

You can select to create a report in one of the following formats, which you choose from the **Output format** list:

- **HTML**, for creating the report as an HTML file for display in a web browser.
- **Microsoft Word**, for creating the report as a docx file for use as a document in Microsoft Word (version 2007 or later).

When adding a new report, the initial setting for the output format is that of the last report you wrote (or previewed).

For HTML reports, select the **Open in browser** check box to view the report in a web browser as soon as it is available. Select the **Optimize for printing** check box to generate images with resolution adapted for printing rather than screen display.

For reports in the docx format, select the **Open finished report** to directly open the generated report in Microsoft Word.

The Report Generator stores the report in a file with the chosen name and by default gives it the extension `.html` or `.docx`. For HTML reports, it stores images included in the report and the style sheet in a subdirectory with the same name as the report plus the suffix `_files`. Reports in Microsoft Word format are self contained.

When generating a report, you need to specify its name and where to store the file. First, use the **Report filename** list to specify whether to link the report name to the name of the model's MPH-file (**From model**) or to specify it independently (**Custom**).

If **From model** is selected, enter the directory for the report in the **Report directory** field or click **Browse** to launch the **Specify Report Directory** dialog box and browse to the desired location. The default directory is the directory where a report was last saved.

If, instead, **Custom** is selected from the **Report filename** list, enter the path and filename in the **Filename** field or click **Browse** to launch the **Specify Report File** dialog box, browse to the desired location, and enter the filename in the **File name** field.

Publishing and Editing an HTML Report

Once an HTML report is created, it can be published or edited in any HTML editor. If you want to publish a report on the Internet or send it to a colleague, also send the HTML file as well as the folder with the images and style sheet.

IMAGES

Select the **Disable image generation** check box if you want to quickly rewrite the report to include changes that only affect the text. If the report includes images that are not available, COMSOL overrides this setting and generates new images. From the **Size** list, you can select the image size as **Extra small** (240-by-180 pixels for HTML screen output/2.4-by-1.8 inches for print-optimized output), **Small** (320-by-240 pixels/3-by-2.4 inches), **Medium** (480-by-360 pixels/4-by-3 inches), or **Large** (600-by-450 pixels/6-by-4.5 inches). All images have an aspect ratio of 4:3. From the **Type** list, select the image file type: **PNG** (the default), **JPEG**, or **BMP** (not supported for reports in Microsoft Word format; choosing this option gives PNG images).

For the best image quality, use PNG images.

Edit the Default Report Settings

Open The Preferences Dialog Box and click **Results** to control some properties for the report generator.

- In the **Default report style-sheet file** field type the full path and filename to a style sheet (.css-file) that you want to use as the default style sheet for reports in HTML format. Click **Browse** to browse to a file that contains the style sheet.
- In the **Logotype file** field, type the full path and filename for an image file (on PNG or JPEG format) to use as the logotype. Click **Browse** to browse to the logotype file. If empty, the COMSOL logotype appears in the report.
- In the **Company** field, type the name of a company associated with the report if desired.
- From the **Default image size** list you can select the default size for report images; choose between **Extra small**, **Small**, **Medium**, and **Large**. Similarly, use the **Default image type** list to specify the default type for the report images: **PNG**, **JPEG**, or **BMP**.
- Select the **Prompt for update of table of contents in Microsoft Word** check box to make Microsoft Word ask whether you want to update the table of contents when you first launch a report in this format. Such an update is necessary to generate page numbers in the table of contents, but you can choose to do the update after you have opened the document in Microsoft Word. By default this check box is not selected.

The Documentation Node

The main **Documentation** node ( ) contains information about the formatting and defaults for a documentation. Click the **Preview Selected** ( ) or **Preview All** ( ) button to show a preview of the document in the **Preview** window. Click the **Write** button ( ) in the toolbar for the **Documentation** settings window to create a document. The **Write** option is also available by right-clicking any node in the documentation. Selecting **Write** from any documentation node's context menu generates the entire document.

FORMAT

You can select to create a document in one of the following formats, which you choose from the **Output format** list:

- **HTML** (the default format), for creating the document as an HTML file for display in a web browser.
- **Help plug-in**, for creating the document as a plug-in of Eclipse Help format that can become an integral part of the documentation and help system.

Settings for Documentation in HTML Format

When generating documentation, you need to specify its name and title where to store the file. Enter the output directory for the HTML files in the **HTML output directory** field, or click **Browse** to open the **Specify HTML Output Directory** dialog box and browse to the desired location. Also specify the document's name a title in the **Document name** and **Document title** fields, respectively.

Settings for Documentation as Help Plug-in

When creating a help plug-in, you need to specify the following plug-in location details and properties:

- Enter the output directory for the help plug-in in the **Plug-in output directory** field, or click **Browse** to open the **Specify Plug-in Output Directory** dialog box and browse to the desired location.
- Enter a **Plug-in prefix** to use a common name space for your help plug-ins. You can specify a standard plug-in prefix for your organization on the **Builder Tools** page in the **Preferences** dialog box.
- Enter the plug-in name and title in the **Plug-in name** and **Plug-in title** fields, respectively. The complete plug-in name is formed from the plug-in prefix and plug-in name. The plug-in title is the title that appears in the **Contents** tree on the left side of the standalone **Help** window or on the **Contents** page of the **Help** window for context help integrated in the COMSOL Desktop.
- From the **Add to** list, select **None** (the default) to not link the plug-in to the contents of any COMSOL product, or select COMSOL Multiphysics or any of its modules or LiveLink products to add the documentation to one of the products. As an advanced option, you can select **Custom** to link to a custom plug-in that you specify in the **Link to plug-in named** field and the **Link to anchor ID** field. The custom plug-in must extend the org.eclipse.help.toc extension point and contain the specified anchor for your documentation plug-in to link to.
- In the **Vendor** field you can enter the name of your organization. This name appears in the manifest file inside the generated plug-in.
- Enter the version of the plug-in in the **Plug-in version** field. The default version is 1.0.0. Like the vendor name, this number is written to the plug-in's manifest file.

The Title Page

By default, all nonempty reports start with a title page (if it is not needed, right-click the applicable node and choose **Delete**). The **Title Page** node (▤) defines general settings and information about the model. The default name is the model's filename.

FRONT MATTER

In the **Report title** list, choose between linking the report's title to the MPH-file's name (**From model**) or specifying it independently (**Custom**). If you choose **Custom**, use the **Title** field to give the title of the report (the default is based on the MPH-file).

From the **Image** list, select an option for an image in the report title—**None**, **Model thumbnail** (the default), or any of the plot groups' plots in the model. See Setting the Model Thumbnail Image for a Model for information about the thumbnail.

From the **Layout** list, select **Table** (the default) to present the model settings such as author and date in a table, or select **Headings** to present these settings using headers.

If the **Show logotype** check box is selected (the default) the report include a logotype that you have specified in the **Preferences** dialog box or, if no such logotype is available, the COMSOL logotype.

Use the **Author**, **Date**, **Company**, and **Version** fields if desired to provide that information in the report. Clear the check box in front of each setting to exclude it from the report.

The **Summary** text box contains the comments from the model's **Root** node (the model description) by default. Clear the associated check box to exclude the summary from the report.

The **Acknowledgment** text box is empty by default. Clear the associated check box to exclude the acknowledgment from the report.

The Table of Contents

The **Table of Contents** node (▤) contains the table of content for the report.

LEVELS

The **Section levels in table of contents** list determines how many section levels to include in the table of contents: 1–5 (default: 2).

TABLE OF CONTENTS

This section contains the current table of contents.

Sections in the Report

The **Section** nodes (☰) provide the structure of the report. You can add sections in several levels by right-clicking a **Section** node to add additional **Section** nodes as subsections. The **Section** node's context menu also contains two submenus: **Basic**, for adding basic report component such as paragraphs, images, and tables, and **Model**, for adding information about the model such as the geometry, mesh, physics interfaces, and plot groups.

SECTION HEADING

From the **Source** list, select the source of the section's heading:

- **Custom** (the default). You then specify the heading in the **Heading** field.
- **From first child node**. The section heading is the name of the first child node under the **Section** node.

Basic Report Components

Right-click nodes to select and add these report nodes from the **Basic** submenu—**Bibliography**, **Code**, **Equation**, **Heading**, **Image**, **List**, **Note**, **Table**, and **Text**.

The group of report components provide basic building blocks for a report as described in Table 20-12.

TABLE 20-12: BASIC REPORT AND DOCUMENTATION COMPONENTS

ICON	REPORT COMPONENT	DESCRIPTION
☰	Bibliography	Adds a reference or bibliography to the report or document. Right-click to add **Reference** www nodes.
▦	Code	Adds a text block for code using a code (monospace) font. You can also make part of the text using an italic or bold variant of the code font.

TABLE 20-12: BASIC REPORT AND DOCUMENTATION COMPONENTS

ICON	REPORT COMPONENT	DESCRIPTION
∫du	Equation	Adds an equation to the report or document. You can use LaTeX markup directly or import the equation as an image. Under **Equation preview** you can see the equation that the LaTeX commands that you enter create.
	Heading	Adds a heading to the report or document with a text from the **Text** field and a layout for the level (Level 1–Level 6) from the **Level** list. The default is to use the level where the Heading node appears.
	Image	Adds an image to the report or document. Select the image source from the **Source** list: **Plot group** to select the plot from available plots in the Plot group list or **External** to use any external image file in PNG, Windows Bitmap (BMP), or JPEG format. Add a **Caption** if desired.
	List	Adds a list. By default, the **Numbered** check box is selected, giving a numbered list; clear the check box for an unordered (bullet) list. Right-click the **List** node to add **List Item** nodes.
	List Item	Right-click the **List** node to add this node with a **Text** area for the list item's contents. Right-click to add **Code**, **Equation**, **Image**, **Table**, **Text**, or other **List** nodes for inserted texts, equations, images, or tables in the list or for creating nested lists.
	Note	Adds a **Note** node for adding one of the following note types, which you select from the **Type** list: **Note** (the default), **Caution**, **Important**, **Model**, **See also**, or **Tip**. From the **Show** list, select **Icon** (the default) to display the icon only, **Description** (the type), or **Icon and description**. Then add the text for the note.
www	Reference	Right-click a **Bibliography** node to add references. The reference information that you provide is formatted based on the type of reference that you select from the **Type** list: **Journal article** (the default), **Book**, **Conference paper, Thesis**, or **Web**.
	Table	Adds a table with a **Title** and a **Number of columns** (default: 3 columns). Right-click to add a **Table Heading Row** and **Table Rows**.
	Table Heading Row	Right-click the **Table** node to add this node and then define headings for each column.
	Table Rows	Right-click the **Table** node to add this node and then add the contents for each column in a row of a table.
T	Text	Provides a **Text** area where text can be added (including HTML tags for formatting and links).

For all **Text**, **List Item**, and **Note** nodes' settings, a set of tools above and beyond the text field provides a quick way to add formatting to the text:

- The formatting tools above the text provide character formats for user-interface labels, emphasis, code (standard, bold, and italic), equation components (bold, variables, and constants), subscript, and superscript. To convert a part of the text to any of these character formats, highlight the text that you want to format and then click **L** , for example, to mark the text as a user-interface label (a sans-serif boldface font) using the HTML tags `<l>` and `</l>` before and after the text.

- From the character tools below the text, click the character that you want to insert, for example, click Ω to insert an uppercase omega as `\Omega` in the text. The character tools include lowercase and uppercase Greek letters and the en-dash (–) and em-dash (—) punctuation symbols.

Click **Preview Selected** ( ) to display a preview of the text, including formatting, in the **Preview** window.

Mathematical Symbols and Special Characters

COMSOL supports a subset of the LaTeX language for creating equations as part of the documentation or in user-developed physics interfaces and other applications. Commands include Greek and other characters, mathematical symbols and operators, arrows, text and font formats, and environments for text and mathematical typesetting. The following tables and lists contain the commands that are available for creating equations and other mathematical text.

If the LaTeX syntax is not correct or not included in COMSOL, the equation preview is empty, but no error appears.

GREEK AND OTHER CHARACTERS

The following table contains the supported lowercase and uppercase Greek letters and the Swedish character Å:

TABLE 20-13: GREEK AND OTHER CHARACTERS

COMMAND (UPPERCASE)	CHARACTER	COMMAND (LOWERCASE)	CHARACTER
		\alpha	α
		\beta	β
\Gamma	Γ	\gamma	γ
\Delta	Δ	\delta	δ
		\varepsilon	ε
		\epsilon	ϵ
		\zeta	ζ
		\eta	η
\Theta	Θ	\theta	θ
		\vartheta	ϑ
		\iota	ι
		\kappa	κ
\Lambda	Λ	\lambda	λ
		\mu	μ
		\nu	ν
\Xi	Ξ	\xi	ξ
\Pi	Π	\pi	π
		\varpi	ϖ
		\rho	ρ
\Sigma	Σ	\sigma	σ
		\varsigma	ς
		\tau	τ

TABLE 20-13: GREEK AND OTHER CHARACTERS

COMMAND (UPPERCASE)	CHARACTER	COMMAND (LOWERCASE)	CHARACTER
\Upsilon	Υ	\upsilon	υ
\Phi	Φ	\phi	ϕ
		\varphi	φ
		\chi	χ
\Psi	Ψ	\psi	ψ
\Omega	Ω	\omega	ω
\AA	Å		

ACCENTS

The following accents are available:

TABLE 20-14: GREEK AND OTHER CHARACTERS

COMMAND	ACCENT	COMMAND	ACCENT
\acute	$\acute{e}$	\bar	$\bar{e}$
\breve	$\breve{e}$	\check	$\check{e}$
\ddot	$\ddot{e}$	\dot	$\dot{e}$
\grave	$\grave{e}$	\hat	$\hat{e}$
\tilde	$\tilde{e}$	\vec	$\vec{e}$

MATHEMATICAL SYMBOLS AND OPERATORS

The following mathematical symbols and operators are available:

TABLE 20-15: GENERAL SYMBOLS AND MATHEMATICAL OPERATORS

COMMAND	SYMBOL	COMMAND	SYMBOL
\dots	…	\nabla	∇
\ldots	…	\bot	⊥
\hbar	▯	\diamondsuit	♦
\Re	ℜ	\neg	¬
\forall	∀	\lnot	¬
\cdots	⋯	\imath	ı
\Im	ℑ	\exists	∃
\prime	′	\triangle	Δ
\top	⊤	\heartsuit	♥
\flat	♭	\vdots	⋮
\Diamond	◇	\aleph	ℵ
\mho	℧	\emptyset	∅

TABLE 20-15: GENERAL SYMBOLS AND MATHEMATICAL OPERATORS

COMMAND	SYMBOL	COMMAND	SYMBOL
\infty	∞	\angle	∠
\clubsuit	♣	\pounds	£
\ddots	⋱	\Box	□
\wp	℘	\partial	∂
\surd	√	\spadesuit	♠
\dag	†	\ddag	‡
\S	§	\P	¶
\copyright	©	\textregistered	®

The following table lists the available "big" mathematical operator as well as binary mathematical operators and relations:

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\sum	∑	\prod	∏
\coprod	∐	\int	∫
\bigoplus	⨁	\bigcup	⋃
\bigcap	⋂	\bigsqcup	⨆
\oint	∮	\bigotimes	⨂
\bigvee	⋁	\bigwedge	⋀
\biguplus	⨄	\bigodot	⨀
\pm	±	\cdot	·
\times	×	\cup	∪
\sqcup	⊔	\vee	∨
\lor	∨	\oplus	⊕
\otimes	⊗	\lhd	⊲

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\unlhd	⊴	\mp	∓
\div	÷	\setminus	\
\cap	∩	\sqcap	⊓
\wedge	∧	\land	∧
\ominus	⊖	\oslash	⊘
\rhd	⊳	\unrhd	⊵
\star	★	\ast	*
\circ	∘	\bullet	•
\uplus	⊎	\amalg	∐
\dagger	†	\ddagger	‡
\wr	≀	\leq	≤
\le	≤	\ll	>>
\prec	≺	\preceq	≼
\subset	⊂	\subseteq	⊆
\sqsubset	⊏	\sqsubseteq	⊑
\in	∈	\vdash	⊢
\mid	|	\geq	≥
\ge	≥	\gg	>>
\succ	≻	\succeq	≽
\supset	⊃	\supseteq	⊇
\sqsupset	⊐	\sqsupseteq	⊒
\ni	∋	\owns	∋
\dashv	–	\parallel	||
\notin	∉	\equiv	≡
\doteq	≐	\sim	∼
\simeq	≃	\approx	≈
\cong	≌	\bowtie or \Join	⋈
\propto	∝	\models	⊨

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\perp	⊥	\asymp	≍
\neq	≠	\ne	≠
\bigtriangleup	△	\bigcirc	○
\bigtriangledown	▽	\triangleleft	◁
\triangleright	▷	\diamond	⋄
\smile	⌣	\frown	⌢

ARROWS

The following table includes the available types of arrows:

TABLE 20-17: ARROWS

COMMAND	SYMBOL	COMMAND	ARROW
\leftarrow or \gets	←	\rightarrow or \to	→
\Leftarrow	⇐	\Rightarrow	⇒
\Leftrightarrow	⇔	\leftrightarrow	↔
\hookleftarrow	↩	\leftharpoonup	↼
\leftharpoondown	↽	\leftrightharpoons	⇋
\nearrow	↗	\swarrow	↙
\leadsto	↝	\hookrightarrow	↪
\rightharpoonup	⇀	\rightharpoondown	⇁
\searrow	↘	\nwarrow	↖
\downarrowtobar	⤓	\uparrowtobar	⤒
\downtrianglefilled	▼	\uptrianglefilled	▲
\mapsto	↦	\longmapsto	⟼
\longleftarrow	⟵	\longrightarrow	⟶
\longleftrightarrow	⟷	\Longleftarrow	⟸
\Longrightarrow	⟹	\Longleftrightarrow	⟺
\iff	⇔		

DELIMITERS AND ENVIRONMENTS

The following tables includes the available delimiter, spaces, and environments:

TABLE 20-18: DELIMITERS, SPACES, ENVIRONMENTS, BOXES

COMMAND	SYMBOL/EXPLANATION	COMMAND	SYMBOL/EXPLANATION
\uparrow	↑	\updownarrow	↕
\downarrow	↓	\Uparrow	⇑
\Updownarrow	⇕	\Downarrow	⇓
\lbrack	[	\rbrack	]
\lbrace	{	\rbrace	}
\vert	|	\backslash	\
\Vert	‖	\lfloor	⌊
\lceil	⌈	\rfloor	⌋
\rceil	⌉	\langle	⟨
\rangle	⟩	\left	Delimiter sizing (see Note below)
\right	Delimiter sizing (see Note below)	\quad	Explicit horizontal spacing
\qquad	Double explicit horizontal spacing	\raisebox	Creates a box containing text; it is used to raise or lower text.
\mbox	Enclose text in a box	\phantom	Adds an invisible component to, for example, balance subscripts
\begin	Invoke the *array* environment (see Note below)	\end	End the *array* environment (see Note below)
\unicode	Display Unicode characters as supported by the font: \unicode{ÅÄÖ}, for example	\hspace	Horizontal space

The `\left` and `\right` commands must be used in pairs to provide flexible delimiters that fit the formula inside. Put the desired delimiter—(and), for example—immediately after the `\left` and `\right` commands. For example, `\left( \frac{x}{y} \right)` provides x/y as a fraction within parentheses that fit the expression's size.

The `\begin` and `\end` commands must be used in pairs to mark the beginning and end of an environment. The only supported environment is the `array`. For example, `\begin{array}{clcr} 1 & 2 & 3 \\ 4 & 5 & 6\end{array}` creates a matrix with 2 rows and three columns.

MATHEMATICAL FUNCTION NAMES

The following function commands provide the function name using a Roman font:

TABLE 20-19: FUNCTIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\arccos	arccos	\cos	cos
\csc	csc	\exp	exp
\ker	ker	\limsup	limsup
\arcsin	arcsin	\cosh	cosh
\deg	deg	\gcd	gcd
\lg	lg	\ln	ln
\arctan	arctan	\cot	cot
\det	det	\hom	hom
\lim	lim	\log	log
\arg	arg	\coth	coth
\dim	dim	\inf	inf
\liminf	liminf	\max	max
\sinh	sinh	\sup	sup
\tan	tan	\tanh	tanh
\min	min	\Pr	Pr
\sec	sec	\sin	sin

SPECIAL MATHEMATICAL TYPESETTING

There are two mathematical formula components with a special syntax: `\frac` for fractions and `\sqrt` for roots:

- Use the syntax `\frac{`*`numerator`*`}{`*`denominator`*`}` to create a fraction. For example, the expression `\frac{n!}{k!(n-k)!}` produces the following output:

$$\frac{n!}{k!(n-k)!}$$

- Use the syntax `\sqrt[`*`order`*`]{`*`expression`*`}` to create a root surrounding an expression. The `[`*`order`*`]` argument is optional; without it, the syntax produces a square root. For example, `\sqrt[n]{1+x^2}` produces the following output:

$$\sqrt[n]{1+x^2}$$

TEXT AND FONT ELEMENTS

The following syntax elements are available for creating different text elements and fonts:

TABLE 20-20: VARIOUS TEXT AND FONT OPERATIONS

COMMAND	EXPLANATION	COMMAND	EXPLANATION
\textsuperscript	Superscripts	^	Superscripts
\textsubscript	Subscripts	_	Subscripts
\overline	Overlining	\underline	Underlining
\overleftarrow	Overlining using a left-pointing arrow	\underleftarrow	Underlining using a left-pointing arrow
\overrightarrow	Overlining using a right-pointing arrow	\underrightarrow	Underlining using a right-pointing arrow

TABLE 20-20: VARIOUS TEXT AND FONT OPERATIONS

COMMAND	EXPLANATION	COMMAND	EXPLANATION
\overbrace	Overlining using a brace	\underbrace	Underlining using a brace
\textnormal	Normal text	\textbf	Boldface text
\textit	Text in italics	\textrm	Text in Roman font
\mathnormal	Normal mathematical mode (the default)	\mathbf	Mathematical boldface text
\mathit	Mathematical text in italics	\mathrm	Mathematical text in Roman font
\displaystyle	Size for equations in display mode	\textstyle	Size for equations in text mode
\scriptstyle	Size for first subscript or superscript	\scriptscriptstyle	Size for subsequent subscripts and superscripts
\emph	Emphasize text in normal (Roman) text mode	\tiny	Smallest font size in text mode
\scriptsize	Second smallest fontsize in text mode	\footnotesize	Third smallest font size in text mode
\small	Fourth smallest fontsize in text mode	\normalsize	Normal font size in text mode
\large	Fifth largest font size in text mode	\Large	Fourth largest font size in text mode
\LARGE	Third largest font size in text mode	\huge	Second largest font size in text mode
\Huge	Largest font size in text mode		

The `\textsuperscript` and ^ syntax alternatives are identical for creating superscripts. Likewise, `\textsubscript` and _ are identical for creating subscripts.

SPECIAL CONTROL SEQUENCES AND CHARACTERS

The following special control sequences and special characters are available:

TABLE 20-21: SPECIAL CONTROL SEQUENCES AND SPECIAL CHARACTERS

<table>
<tr><th>SEQUENCE/CHARACTER</th><th>DESCRIPTION</th><th>SEQUENCE/CHARACTER</th><th>DESCRIPTION</th></tr>
<tr><td>\#</td><td># character</td><td>\:</td><td>Medium space</td></tr>
<tr><td>\$</td><td>$ character</td><td>\;</td><td>Thick space</td></tr>
<tr><td>\%</td><td>% character</td><td>\!</td><td>Negative thin space</td></tr>
<tr><td>\&</td><td>& character</td><td>$, or /[to start and /] to end</td><td>Start and end mathematical mode in text mode</td></tr>
<tr><td>_</td><td>_ character</td><td>%</td><td>Insert comments</td></tr>
<tr><td>\{</td><td>{ character</td><td>&</td><td>Separate items in arrays</td></tr>
<tr><td>\}</td><td>} character</td><td>~</td><td>Nonbreaking space</td></tr>
<tr><td>\|</td><td>|| character</td><td>_</td><td>Subscript</td></tr>
</table>

TABLE 20-21: SPECIAL CONTROL SEQUENCES AND SPECIAL CHARACTERS

SEQUENCE/CHARACTER	DESCRIPTION	SEQUENCE/CHARACTER	DESCRIPTION
\<*space*>	Space	^	Superscript
\,	Thin space	\, {, }, [,]	Command syntax elements

Model Report Components

Right-click nodes to select items from the **Model** submenu. This group of report components provide information about the model are detailed in these sections:

- Definitions Report Nodes
- Results Report Nodes
- Geometry Report Node
- Material Report Node
- Mesh Report Node
- Component Report Node
- Physics Interface Report Node
- Root Report Node
- Solver Report Node
- Study Report Node

Definitions Report Nodes

Right-click nodes to select items from the **Model** submenu and to add the nodes described in this section. These report components provide information about nodes added to the model under Definitions and the Global Definitions node (for variables).

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

PARAMETERS REPORT NODE

Use the **Parameters** report node (P_i), selected from the **Model** submenu, to include the Global Parameters in the model.

COORDINATE SYSTEM

Use the **Coordinate System** report node ( ), selected from the **Model** submenu, to add coordinate system settings information to the report. Select a **Coordinate system** from the list. Select the **Include settings** check box as required.

FUNCTION

Use the **Function** report node (f(x)), selected from the **Model** submenu, to add a function image and the function's settings to the report. Under **Referenced Function**, select a **Function** from the list. The **Include image** check box is selected by default. Select the **Include settings** check box as required.

INFINITE ELEMENT DOMAIN

Use the **Infinite Element Domain** report node (∞), selected from the **Model** submenu, to include the selection and settings for an infinite element domain if it is present in the model. From the **Source** list, select the Infinite Element Domain node to report. Select the **Include settings** check box to include the infinite element settings.

COMPONENT COUPLING

Use the **Component Coupling** report node ( ), selected from the **Model** submenu, to include the settings and a selection image for a component coupling if it is present in the model. Select the **Component coupling** from the list.

The **Include settings** check box is selected by default if there is a component coupling. Select the **Include selection image** check box as required. The image shows the selection for the component coupling and is not present if the component coupling has no selection.

PAIR

Use the **Pair** report node ( ), selected from the **Model** submenu, to include the settings and a selection image for identity pairs and contact pairs if present in the model. Select the **Pair** from the list. The **Include settings** check box is selected by default if there is a pair. Select the **Include selection image** check box as required. The image shows the selection for the pair and is not present if the pair has no selection.

PERFECTLY MATCHED LAYER

Use the **Perfectly Matched Layer** report node ( ), selected from the **Model** submenu, to include the selection and settings for a perfectly matched layer (PML) if it is present in the model. From the **Source** list, select the **Perfectly Matched Layer** node to report. Select the **Include settings** check box to include the PML settings. The image shows the selection for the PML and is not present if the PML has no selection.

PROBE

Use the **Probe** report node ( ), selected from the **Model** submenu, to includes the settings and a selection image for a Probe if it is included in the model. Select the **Probe** from the list. The **Include settings** check box is selected by default if there is a probe. Select the **Include selection image** check box as required. The image shows the selection for the probe and is not present if the probe has no selection.

SELECTION

Use the **Selection** report node ( ), selected from the **Model** submenu, to includes the settings and a selection image for selections. Select the **Selection** from the list. The **Include settings** check box is selected by default if there is a selection. Select the **Include selection image** check box as required. The image is not present if the selection is empty.

VARIABLES

Use the **Variables** report node ( ), selected from the **Model** submenu, to include global or local variable settings. Select the **Variables** from the list.

Results Report Nodes

Right-click to select items from the **Model** submenu and to add the nodes described in this section. These report components provide information about nodes added to the model under Results. For any of these click the **Go to Source** button ( ) to move to the source node under the applicable node under **Results**.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

DATA SET

Use the **Data Set** report node ( ), selected from the **Model** submenu, to include the settings and a selection image for the referenced Data set. Select a **Data set** from the list or click the **Data Sets**. The **Include settings** check box is selected by default. Select the **Include selection image** check box as required.

DERIVED VALUES

Use the **Derived Values** report node ( ), selected from the **Model** submenu, to add derived values settings information to the report. Select a **Derived values** from the list. The **Include settings** check box is selected by default.

EXPORT

Use the **Export** report node ( ), selected from the **Model** submenu, to add any of the images or animations added to the model Export branch. From the **Object** list, select from any available image or animation objects to include in the report (or select **None**).

> Animations are not supported for reports in Microsoft Word format; if included, they are silently ignored when writing the report.

After selecting the **Object**, select an option from the **Size** list—**Object setting** to use the width and height specified in the image or animation settings, or **Report image type** to use the image format specified in the report's root node settings.

Select an option from the **File format** list—**Object setting** to use the file format for the selected object or **Report image type** to be determined by the image type setting specified in the report's root node.

In the **Caption** field enter text as required. By default, this field is left empty and no caption is included in the report.

PLOT GROUP

Use the **Plot Group** report node ( ), selected from the **Model** submenu, to add plots to the report. Select the **Plot group** from the list. From the **Caption source** list, select **From plot group title** (the default) to use the plot's title as the caption, **Custom** to enter a different **Caption** in the field, or **None** for no caption.

TABLE

Use the **Table** report node ( ), selected from the **Model** submenu, to add the table settings to the report. Select the **Table** from the list.

Geometry Report Node

Use the **Geometry** report node ( ), selected from the **Model** submenu, to add the image, units (length and angular), and statistics to the report for specific geometry features.

> Go to Common Results Node Settings for links to information about this section: **Node Properties**.

REFERENCED GEOMETRY

Select a **Geometry** from the list. The **Include image** and **Include units** check boxes are selected by default. Select the **Include statistics** check box as required.

In the **Features** table under **Name**, all the features used in the geometry sequence are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove feature-specific settings, in the **Settings** column, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Material Report Node

Use the **Material** report node ( ), selected from the **Model** submenu, to add the image, selection, and settings to the report for the material property groups in the material.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

MATERIAL

Select a **Material** from the list. The **Include image** and **Include selection** check boxes are selected by default. Select the **Include settings** check box as required. The image shows the selection for the material and is not present if the material has no selection or is completely overridden by other materials' selections.

In the **Features** table under **Name**, all the material properties used in the material are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove material properties settings, in the **Settings** and **Functions** columns, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Mesh Report Node

Use the **Mesh** report node ( ), selected from the **Model** submenu, to add the image and statistics to the report.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

MESH

Select a **Mesh** from the list. The **Include image** check box is selected by default. Select the **Include statistics** check box as required.

In the **Features** table under **Name**, all the mesh features are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove settings, in the **Settings** column, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Component Report Node

Use the **Component** report node ( ), selected from the **Model** submenu, to include information from a Component node in the model.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

COMPONENT SETTINGS

Select the **Component** from the list. Select the **Include unit system** and **Include geometry shape order** check boxes as required.

Physics Interface Report Node

Use the **Physics Interface** report node ( ), selected from the **Model** submenu, to add an image and table showing the selection, equations, settings, and a table of all included physics features to the report.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

REFERENCED INTERFACE

Select a **Physics interface** from the list. The **Include selection image** and **Include feature table** check boxes are selected by default. Select the **Include selection table**, **Include equations**, and **Include settings** check boxes as required. The image shows the selection for the physics and is not present if the physics has no selection.

In the **Features** table under **Name**, all the physics interface features are listed. By default, all settings and selections are included in **Intermediate** and **Complete** reports, and no settings or selections are included for a **Brief** report.

To add or remove settings and selections, in the **Settings** and **Selection** columns, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Under the **Features** table, you can also select from the check boxes to include variables, shape functions, weak expressions, and constraints contained in the physics node's **Equation View** subnode. By default, a **Complete** report includes all of these items and the **Intermediate** reports include variables. **Brief** reports do not include any items.

Select the **Include variables**, **Include shape functions**, **Include weak expressions**, and **Include constraints** check boxes as required.

Root Report Node

Use the **Root** report node ( ), selected from the **Model** submenu, to include information from the model's Root node—model name and path, COMSOL program version, the used products, and unit system.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

GLOBAL SETTINGS

Select the **Include name**, **Include path**, **Include program**, **Include unit system**, and **Include used products** check boxes as required.

Solver Report Node

Use the **Solver** report node ( ), selected from the **Model** submenu, to add settings for the solver nodes in the solver sequence that is referenced in the Sequence list.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

SOLVER

Select a **Sequence** from the list. Select the **Include log** to include the solver log.

In the **Features** table under **Name**, all the solver features are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove settings, in the **Settings** column, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Study Report Node

Use the **Study** report node ( ), selected from the **Model** submenu, to add settings for a Study in the model to the report.

Go to Common Results Node Settings for links to information about this section: **Node Properties**.

STUDY

Select a **Study** from the list.

In the **Features** table under **Name**, all the studies are listed. By default, all settings are included for all report types. To add or remove settings, in the **Settings** column, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✕) to remove a feature from the report.

Printing and Capturing Screen Shots

Printing from the COMSOL Desktop

To print the contents in the **Graphics** window or other plot windows, click the **Print** button (🖨) on the Graphics window toolbar or press Ctrl+P.

In the **Print** dialog box, follow these steps:

1. Under **Image**, from the **Size** list select **Manual** (the default) to define the print size manually using the settings below, or select **Current** to use the current size of the **Graphics** window. Of the settings below, only the **Antialiasing** check box is then available.
2. Select a **Unit** to define the image size—**Millimeters (mm)**, **Inches (in)**, or **Pixels (px)**.
3. Select the **Lock aspect ratio** check box to maintain the calculation of the width and height (if one or the other is changed).
4. Enter the **Width** and **Height** in the units selected for the image.
5. Enter the **Resolution** for the image in DPI (dots per inch) as a value between 10 and 1200 DPI. The default value is 300 DPI.

   Under these settings, the dialog box shows the resulting image size and size on the screen in pixels.
6. The **Antialiasing** check box is cleared by default. Click to select if required. Antialiasing minimizes distortion such as jagged edges in the image.
7. Under **Layout**, the **Title**, **Legend**, **Axes**, and **Logotype** check boxes are selected by default to display the information on the screen shot if you select the **Include** check box. You can then also edit the selections for including these parts of the plot.
8. Enter a **Font** size in points (pt) as a number between 1 and 1000 pt. The default value is 9 pt.
9. Select a **Background**—**Current** or **Color**. Current is the background color in the plot window on the COMSOL Desktop. If **Color** is chosen, click the **Color** button to select a custom color from the color palette that opens.
10. Click **OK** to print the contents of the plot window. Typically the operating system's **Print** dialog box first opens for selecting a printer, the number of copies, and other printer settings.

For generating snapshots of a plot window to a file or the clipboard, click the **Image Snapshot** button (📷) on the Graphics toolbar.

Capturing and Copying Screen Shots

To quickly capture a screenshot image of a plot, press Ctrl+C when the Graphics window or another plot window has focus. The screenshot image is then available on the clipboard so that you can paste it into, for example, a document. Also use the **Image Snapshot** button (📷) on the Graphics toolbar to capture an image snapshot of a plot. To do so, follow these steps:

1. In the **Graphics** window or any other plot window, click the **Image Snapshot** button (📷) to open the **Image Snapshot** dialog box.
2. Under **Image**, from the **Size** list select **Manual** (the default) to define the image size manually using the settings below, or select **Current** to use the current size of the **Graphics** window. Of the settings below, only the **Antialiasing** check box is then available.
3. Select a **Unit** to define the image size—**Millimeters (mm)**, **Inches (in)**, or **Pixels (px)** (the default).
4. Select the **Lock aspect ratio** check box to maintain the calculation of the width and height (if one or the other is changed).

5 Enter the **Width** and **Height** in the units selected for the image.

6 Enter the **Resolution** for the image in DPI (dots per inch) as a value between 10 and 1200 DPI. The default value is 96 DPI.

7 The **Antialiasing** check box is selected by default. Click to clear if required. Antialiasing minimizes distortion such as jagged edges in the image.

8 Under **Layout**, the **Title**, **Legend** (1D graphs) or **Color legend** (2D plots), **Axes**, and **Logotype** check boxes (1D and 2D images) or the **Title**, **Color legend**, **Grid**, **Axis orientation**, and **Logotype** check boxes (3D images) are selected by default to display the information on the screenshot if you select the **Include** check box. You can then also edit the selections for including or excluding these parts of the plot.

9 Enter a **Font** size in points (pt) as a value between 1 and 1000 pt. The default value is 9 pt. This font size overrides the system font size used in the COMSOL Desktop.

10 Select a **Background**—**Current**, **Color**, or **Transparent**. Current is the background color in the plot window on the COMSOL Desktop. If **Color** is chosen, click the **Color** button to select a custom color from the color palette that opens. The **Transparent** option is only available for the PNG file format..

Transparent image support includes two parts: Raw data of the png image and an external renderer (image viewer). COMSOL can provide a correct png image itself but cannot control the external renderer.

When importing an image with a transparent background to another Windows application, first save the image as a file rather than saving it to the clipboard. In some cases the transparent background is not preserved if you copy an image via the clipboard.

11 Under **Output**, select the target: **Clipboard** (the default) copies the image to the clipboard. **File** saves the image to a file.

12 If **File** is selected:

a Select a file **Format**—**BMP**, **EPS** (1D plots only), **JPEG**, or **PNG** (the default).

b Enter a file path in the **Filename** field, or click **Browse** to specify the name and location of the file.

13 Click **OK** to generate the image snapshot.

Printing from the COMSOL Desktop

Setting the Model Thumbnail Image for a Model

To illustrate the model you can save a model thumbnail image that displays in The Root Window and when opening a model in the **Model Libraries** window. The model thumbnail is a copy of the current plot.

To set the model thumbnail:

1 Decide which plot to use as the model thumbnail. In the **Model Builder** under **Results** click the plot group of the plot you want to use so that it displays in the **Graphics** window.

2 Click the **Root** node (the first node in the model tree).

3 On the **Root** settings window under **Model Thumbnail**, click **Set Model Thumbnail**. Save the model file to update the image.

If required, make adjustments to the image in the Graphics window using the toolbar buttons until the image is one that is suitable to your purposes, or choose another plot and repeat the steps.

21

Running COMSOL

This chapter provides an overview of the different ways that you can run the COMSOL Multiphysics® software in addition to running the COMSOL Desktop® graphical user interface on a dedicated computer, including client/server and distributed-memory architectures. For information about using COMSOL with the Amazon Elastic Compute Cloud™ for cloud-based computing, see the separate document *Running COMSOL on the Amazon Cloud*, which is available as a PDF file in the online documentation and on the Windows® start menu.

Running COMSOL

The primary way to access the COMSOL Multiphysics functionality is through the COMSOL Desktop. This section describes alternative means of accessing the functionality in COMSOL, such as running COMSOL in batch mode and in different client/server configurations.

Windows and the Cross-Platform Desktop

COMSOL has a Windows® graphical user interface that is started by default. There is also a cross-platform graphical user interface, which is the same on Windows, Linux, and Mac. You can start the cross-platform interface on Window by double-clicking the file `comsolxpl.exe` in on of the folders `bin\win32` (32 bit) and `bin\win64` (64 bit) in the COMSOL installation directory. The cross-platform interface is the only one available on Linux and Mac.

COMSOL Client/Server

The COMSOL Multiphysics client/server architecture lets you access the COMSOL server—COMSOL's computational engine—as a separate process. The COMSOL server is a single user server allowing multiple connections by the same user.

You must have a floating network license (FNL) to run a COMSOL server and a COMSOL client on separate computers. Any valid COMSOL license is sufficient to run a client and the server on the same computer.

To start the COMSOL server under windows, just click **COMSOL Multiphysics Server** in the **Client Server** folder under your COMSOL installation on the start menu. On Linux and Mac, type the command `comsol server` to start the COMSOL server.

For more options for starting the COMSOL server, see the section The COMSOL Commands for your platform. Also see the section Running COMSOL Client/Server for detailed general information about client/server options.

The license server is not the same as a COMSOL server. The license manager can run on a computer different from both the ones used by COMSOL Desktop and COMSOL server.

Parallel COMSOL

COMSOL Multiphysics supports two mutual modes of parallel operation: shared-memory parallel operations and distributed-memory parallel operations, including cluster support.

SHARED-MEMORY PARALLEL MODE

The shared-memory parallel mode is suitable for running COMSOL on modern multicore or multiprocessor computers. This parallel mode of operation is available for all platforms and all license types. By default, COMSOL uses the shared memory parallel mode and allocates all cores on the computer.

For options for controlling the number of cores used and other options, see the section The COMSOL Commands for your platform. Also see the section Shared-Memory Parallel Mode for detailed general information.

DISTRIBUTED-MEMORY PARALLEL MODE

The distributed-memory parallel mode lets you run COMSOL on a Windows HPC cluster or a Linux cluster.

See Distributed-Memory Parallel Mode for details about running COMSOL in parallel architectures, including Windows and Linux clusters. For more options on how to run COMSOL on a cluster from the command line, see the section The COMSOL Commands for your platform.

LiveLink for MATLAB

The LiveLink™ *for* MATLAB® provides access to COMSOL Multiphysics from MATLAB. From the MATLAB prompt you access COMSOL models through a client/server connection to a COMSOL server. You access the model through the COMSOL API and its Java® interface in MATLAB. In addition, there are M-file wrapper functions that help you perform tasks such as displaying graphics using MATLAB figure windows or fetching data from the model object. The model can be accessed simultaneously from the COMSOL Desktop running in client mode connected to the same server as MATLAB. See the LiveLink™ *for* MATLAB® manuals for more information.

LiveLink for Excel

The LiveLink™ *for* Excel® provides access to COMSOL Multiphysics from Excel. From an Excel sheet you can access COMSOL model through a client/server connection to a running COMSOL server. The model can be accessed simultaneously from the COMSOL Desktop running in client mode connected to the same server as Excel. See the LiveLink™ *for* Excel® manuals for more information.

COMSOL Batch

The COMSOL Multiphysics batch mode provides a way to run COMSOL without a graphical user interface. The COMSOL batch mode allows you to run both Model MPH-files and compiled model files for Java (class files).

You can control the options for running COMSOL in batch mode from the **Study** node in the **Model Builder**. To enable the Batch feature, click the **Show** button ( ) and select **Advanced Study Options**. Then in the **Model Builder**, right-click a **Study** node and select **Batch**. Also see Batch (Job Configuration).

You can also run COMSOL batch entirely from a command prompt.

In batch mode you can monitor the memory usage reported in the log as lines of the form

```
Memory: RAM/MAXRAM VIRT/MAXVIRT
```

where `RAM` is the current memory usage in MB, and `VIRT` is the current virtual memory usage in MB. The maximum measured usage is reported in `MAXRAM` and `MAXVIRT`, respectively. The log only reports changes to the memory usage. You can also monitor the current progress, which is reported as lines in the log of the form

```
Current Progress: 53%
```

where the percentage indicates the currently estimated progress.

For options for controlling the batch command options, see the section The COMSOL Commands for your platform.

COMSOL API

The COMSOL API is a Java-based programming interface for COMSOL. The COMSOL API can be used for a variety of purposes such as developing standalone applications based on COMSOL functionality or running a model file for Java from the COMSOL Desktop or using the available batch command.

To run a model file for Java from the COMSOL Desktop, compile it using the COMSOL compile command. This command is called `comsolcompile` on Windows® and `comsol compile` on other platforms. The compilation gets you a model class file corresponding to the model file for Java. Launch the model class file by selecting **Open** on the **File** menu, and selecting a **Model Class File** under **File name**.

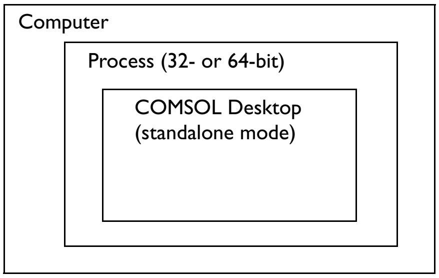
To create a standalone application using the COMSOL API you need to develop a text based or GUI-based interface to the functionality and compile it using the COMSOL compile command. The application can be run in *standalone mode* that links your Java® application directly to the COMSOL code (as a single process). You can also choose to run the application in client/server mode by connecting to a COMSOL server.

To see options for compiling Java files, see The COMSOL Commands for your platform. For more comprehensive information about the COMSOL API, see the *COMSOL API Reference Guide*.

COMSOL Client/Server Architecture

Standalone COMSOL

The most straightforward way of running COMSOL Multiphysics is as a standalone application:

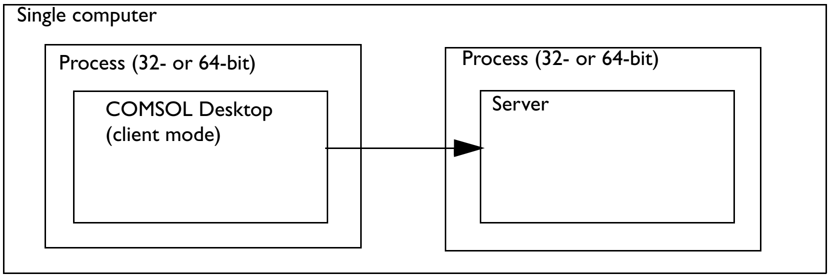


Running COMSOL as a Client/Server

The COMSOL client and server applications are available on all platforms.

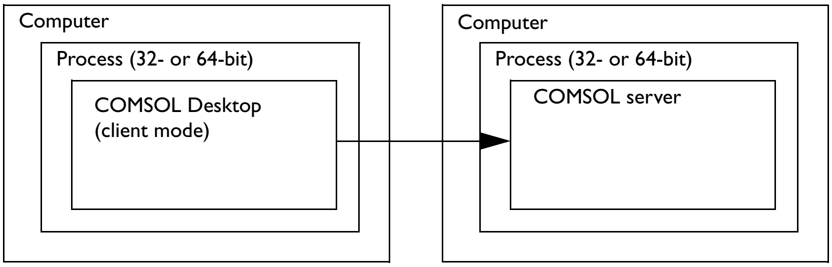
RUNNING COMSOL MULTIPHYSICS CLIENT/SERVER ON THE SAME COMPUTER

Both the COMSOL client and the COMSOL server can run on the same computer and with all available license types: named user license (NSL), CPU locked license (CPU), and floating network license (FNL).



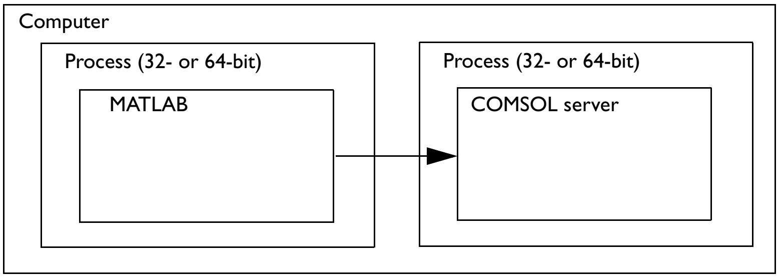
RUNNING COMSOL CLIENT/SERVER ON DIFFERENT COMPUTERS

The COMSOL client and COMSOL Multiphysics server can also run on different computers, but this configuration requires a floating network license (FNL).



Running COMSOL with MATLAB or Excel

COMSOL can run together with MATLAB or Excel on the same machine using the client/server architecture. The command `comsol server matlab` launches this configuration. The command `comsol matlab` can be used to launch just the client and to connect to a remote server. Excel launches a COMSOL server when you open a model.



Running COMSOL Client/Server

The COMSOL Desktop can run in a separate process as a client to a COMSOL Multiphysics server. The COMSOL Desktop client uses a TCP/IP connection to connect to the COMSOL server. The client and server need not run on the same platform. You must have a floating network license (FNL) to run the COMSOL server and the COMSOL client on separate computers.

When you use LiveLink™ *for* MATLAB® and LiveLink™ *for* Excel® a COMSOL client runs within MATLAB and Excel and connects to a COMSOL server, which is typically started automatically.

Advantages of Using COMSOL Client/Server

SOLVING LARGER MODELS

The COMSOL client/server configuration frees your desktop computer of lengthy computations, dispatching your jobs to a dedicated computer. The computer that runs the COMSOL server could have more memory and a faster CPU than your desktop computer.

> Running the COMSOL server and COMSOL client separately on the same computer increases the total memory available to solve problems. This is particularly interesting because the 32-bit limit on addressable memory can be the limiting factor for large models. The COMSOL server components do not use the memory required for the graphical user interface, freeing that memory for the actual computations on the server.

CROSS-PLATFORM CONNECTIONS

The client and server need not run on the same platform. For example, you can run the COMSOL Desktop on Windows® as a client, connecting to a COMSOL server on a Linux or Mac server. In this way you can interactively access a more powerful remote computer.

MULTIPLE CONNECTIONS

Only one graphical user interface can be connected to a COMSOL server at a given time. However, additional clients can be connected to the same server from LiveLink™ *for* MATLAB®and LiveLink™ *for* Excel® and also standalone clients using the COMSOL API. For example, the COMSOL Desktop can act as a COMSOL client when connected to a COMSOL server, and a MATLAB session can be connected to the same server using LiveLink™ *for* MATLAB®.

Running COMSOL Client/Server

STARTING A COMSOL SERVER

- When you have access to the Windows® desktop, start the COMSOL server from the **Start** menu. Go to **All Programs**, select **COMSOL 4.4** and then **Client Server**, and select **COMSOL Multiphysics 4.4 Server**. If starting the COMSOL server from a terminal window in Windows, use the command `<COMSOL installation directory>\bin\win32\comsolserver.exe` for the 32-bit version or `<COMSOL installation directory>\bin\win64\comsolserver.exe` for the 64-bit version.
- On Linux®, use the `comsol server` command to start a COMSOL server.
- On Mac OS X, use the **COMSOL Multiphysics Server** application, or if you connect to Mac OS X from another computer, use the `comsol server` command in the terminal window.

INITIALIZING THE COMSOL SERVER

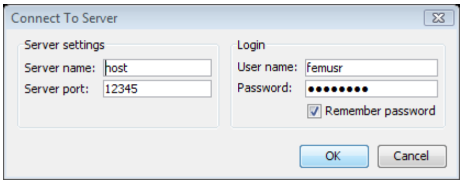
The first time you start a COMSOL server on a computer, you are asked for a username and password. By default, your username and a hashed password is stored on your computer's hard drive. You can avoid storing your username and password on disk by providing the `-passwd nostore` target option to the COMSOL server command. When the COMSOL server is started, the server displays the port number. The server also displays a message each time you log in from a client.

Connecting and Disconnecting from the Desktop

CONNECTING TO A SERVER

To connect to a server from the COMSOL Desktop, select **File>Client Server>Connect to Server** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Connect to Server** ( ) button.

Then in the **Connect to Server** dialog box, specify the hostname of the server, the TCP/IP port number of the server.



Connect To Server

Server settings

Server name: host

Server port: 12345

Login

User name: femusr

Password: ••••••••

☑ Remember password

The port number is displayed by the server as you start it. The port number can change, for example, if you have several COMSOL servers running at the same time on the same computer. Each server is automatically assigned its own separate port number. The username and password are the ones you used when you started the server the first time.

When you connect to a server, you may be asked if you want to save your current model. We recommend that you respond *yes* to get the most current version of your model transferred to the server. If you answer *no*, the latest saved copy of your model is transferred to the server.

When you connect to a server your model is transferred to the server by default. If there is already a model in the server, you may be asked if you want to work with your current model in the desktop or the model on the server.

Windows Toolbars and Menus

DISCONNECTING FROM A SERVER

To close the connection to the server or MATLAB, select **File>Client Server>Disconnect from Server** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Disconnect from Server** ( ) button.

This transfers the current model from the server to the COMSOL Desktop, which no longer runs as a client. The server exits by default but can optionally be started with the option `-multi on` to keep running after the disconnect. When you have used the option `-multi on`, your model is kept in memory in the server, and you can attach to the model from another client later on. If you just close the COMSOL Desktop when connected to a server, your client/server session is automatically disconnected, and the server continues to run with the model in memory if you have specified `-multi on`.

WORKING WITH MATLAB, EXCEL, OR THE COMSOL API

From MATLAB you can use the commands `Modelutil.connect` and `Modelutil.disconnect` to connect and disconnect from a COMSOL server, respectively. The connection to the server is necessary to access and manipulate a model.

From MATLAB (and the COMSOL API) you can create multiple models by using utilities `ModelUtil.create` and `ModelUtil.model` commands.

Import Model from Server

If several models are present it the server, to get access to a particular model, select **File>Client Server>Import Model from Server** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Import Model from Server** ( ) button.

Remove Model from Server

To delete models (remove them from the server) that you have created using `ModelUtil`, select **File>Client Server>Remove Model from Server** ( ). For Windows users, you can also customize the Quick Access Toolbar and then click the **Remove Model from Server** ( ) button.

Windows Toolbars and Menus

Shared Libraries

When running in a Java application, and from MATLAB, the COMSOL client uses Java only and does not load shared libraries. When the COMSOL Desktop operates as a COMSOL client, it loads several shared libraries.

Running COMSOL in Parallel

COMSOL supports two mutual modes of parallel operation:

- The distributed memory model runs on several nodes on a Linux® or Windows® cluster; see Distributed-Memory Parallel COMSOL.
- A parallel shared memory model.

Shared-Memory Parallel COMSOL

Most multiprocessor machines and dual-core/multicore machines support the shared memory model. When running on a cluster, COMSOL uses shared-memory parallelism on each node; and distributed parallelism among the cluster nodes. The solvers, assembly, and meshing in COMSOL Multiphysics benefit from shared-memory parallelism. By default COMSOL uses all cores available on the machine for shared-memory parallelism.

BENEFITS OF RUNNING COMSOL IN SHARED-MEMORY PARALLEL MODE

All iterative solvers and smoothers except Incomplete LU are parallelized. Some smoothers have blocked versions. The blocked versions usually benefit more from running in parallel than the nonblocked versions. The finite element assembly also runs in parallel. Usually the speedup depends on the problem size; problems using a lot of memory usually have better speedup.

The PARDISO sparse direct linear solver runs in parallel. The SPOOLES sparse direct linear solver also runs in parallel. The MUMPS direct solver benefits from shared memory parallelism; however, it does so to a slightly lesser extent than PARDISO and SPOOLES.

The free mesher in 3D runs in parallel over the faces and domains of the geometry object being meshed. For this reason, the speedup when running on several processors depends strongly on the domain partitioning of the corresponding geometry. Meshing a geometry with only one domain, such as an imported CAD part, gives almost no speedup at all. On the other hand, meshing a geometry with several domains, such as an imported CAD assembly with many parts, can give significant speedup, especially if the number of elements in the mesh is large.

These plots run in parallel in 3D: slice plots, isosurface plots, volume plots, line plots, deformed-shape plots, and streamline plots.

These plots run in parallel in 2D: surface plots, contour plots, line plots, and deformed-shape plots.

A significant part of the parallel speedup in computations comes from functions of the BLAS type (basic linear algebra subprogram; see the next section). If you want to run the software in parallel, it is important that the BLAS library you use supports parallelism. The BLAS libraries shipped with COMSOL do that.

Running in parallel usually requires extra memory. If you run out of memory, try to lower the number of used cores as explained in the *COMSOL Installation Guide*. The speedup depends on the processor load. For instance, if your system has m processors and n of them are used by other active programs, do not set the number of processors to a number that is greater than $m - n$. The reason is that the programs compete for the same resources, which slows all of them considerably.

COMSOL and BLAS

BLAS is a set of functions for basic linear algebra operations. Vendors often supply BLAS libraries optimized for their hardware. A large portion of the computational engine in COMSOL relies on BLAS. Included with COMSOL Multiphysics is the MKL (Math Kernel Library) BLAS library, which is the default BLAS library. For AMD processors, COMSOL also includes the ACML (AMD Core Math Library) BLAS library, optimized for AMD processors with SSE2 support, which might improve performance in some cases. It is also possible to supply another BLAS library optimized for your hardware. See the *COMSOL Installation Guide* for information about how to override the default BLAS library (MKL). If the library you want to use is unavailable or incorrectly installed, COMSOL switches to the default BLAS library.

Distributed-Memory Parallel COMSOL

BASIC CLUSTER CONCEPTS

The following terms occur frequently when describing the hardware for cluster computing and shared memory parallel computing:

- Compute node: The *compute nodes* are where the distributed computing occurs. The COMSOL server resides in a compute node and communicates with other compute nodes using MPI (message-passing interface).
- Host: The *host* is a hardware physical machine with a network adapter and unique network address. The host is part of the cluster. It is sometimes referred to as a *physical node*.
- Core: The *core* is a processor core used in shared-memory parallelism by a computational node with multiple processors.

The number of used hosts and the number of computational nodes are usually the same. For some special problem types, like very small problems with many parameters, it might be beneficial to use more than one computational node on one host.

The Linux® and Windows® versions of COMSOL Multiphysics support a distributed memory mode. The distributed mode starts a number of computational nodes set by the user. Each computational node is a separate process running a COMSOL instance. A computational node is not the same as a physical node (computer), but they can coincide. When running in distributed mode, COMSOL uses MPI for communicating between the processes in the distributed environment.

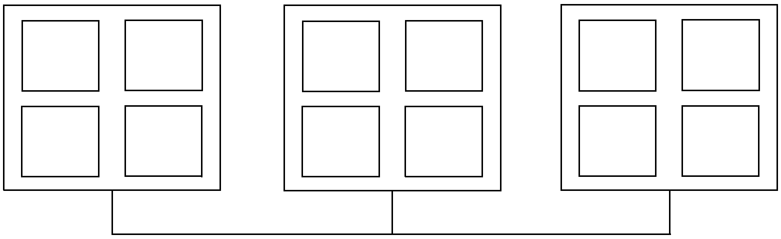


Figure 21-1: Schematic of a cluster with 3 physical nodes (computers) with 4 processors each.

The distributed-memory mode can be combined with the ability of COMSOL to benefit from the shared-memory model. All modes that COMSOL can run in are able to use distributed-memory mode.

In client/server mode, the computer or cluster acting as server must be accessible from the client through a TCP/IP connection. If you are not able to connect to the server, you can use a COMSOL Batch job to solve models on the cluster or do parametric sweeps. You can also use a Cluster Computing study to set up a COMSOL Batch job from within the GUI.

For the schematic in Figure 21-1, you can choose any number of computational nodes between 1 and 12. Each node, in turn, can use between 4 and 1 processors for shared memory. By default, COMSOL uses as many processors as are available on each physical node for shared-memory parallelism on Windows. This is suboptimal if the number of computational nodes is not the same as the number of physical nodes. It is recommended that you explicitly set the number of processors. For the schematic example, if you run 6 computational nodes, the optimal value for number of processors is 2. The number of processors used is $6 \cdot 2 = 12$.

For the same example, assuming you are the sole user of the system for the duration of the computation and that your problem requires a lot of memory, use 3 computational nodes with 4 shared memory cores each. If, on the other hand, your problem is small, use 12 computational nodes with 1 shared memory core each. This way you make the best use of shared-memory and distributed-memory parallelism for each problem.

You do not need a cluster to benefit from the ability to utilize the distributed-memory model. On a multiprocessor computer you can use multiple computational nodes. This can be useful for small-sized parameter sweeps, for example. Make sure that the number of computational nodes times the number of processors does not exceed the number of available processors; otherwise performance deteriorates significantly.

Benefits of Running COMSOL in a Distributed Mode

The following direct solvers are supported by COMSOL Multiphysics when running in distributed mode:

- MUMPS
- SPOOLES

PARDISO is not supported in distributed mode. MUMPS is used instead.

The following iterative solvers are supported:

- Iterative solvers: BiCGStab, CG, GMRES, and FGMRES
- Smoothers and preconditioners: SOR, SOR Gauge, SOR Line, SOR Vector, SCGS, and Vanka
- Multigrid
- Domain decomposition

COMSOL can also run parameter sweeps using the distributed mode. The simplest way to start a distributed parameter sweep is to select the **Distribute parametric sweep** check box in the **Cluster Computing** node's settings window's **Cluster Settings** section. The simplest way to modify an existing model is to add the **Cluster Computing** study and select **Compute** in the study node's settings window.

Running COMSOL in Parallel on Clusters

You control the options for running COMSOL on a cluster from the **Study** node in the **Model Builder**. To enable the cluster computing feature, click the **Show** button ( ) and select **Advanced Study Options**. Then in the **Model Builder**, right-click a **Study** node and select **Cluster Computing** ( ).

You must have a floating network license (FNL) to run COMSOL in distributed-memory parallel mode. See Cluster License Handling for more information.

Cluster Computing

The Micromixer—Cluster Version and Joule Heating of a Microactuator—Distributed Parameter Version models show how to set up a model for running COMSOL in parallel on a cluster: in the first case for faster solution of a large fluid flow model using distributed solver jobs, and in the second case for a distributed parametric sweep.

The following sections describe how to run cluster jobs on Windows and Linux.

RUNNING A CLUSTER JOB ON WINDOWS

This section outlines the main steps for running a cluster job on Windows®. Before you start, check that the installation of COMSOL follows these guidelines:

- Make sure that the COMSOL installation directory is shared between all the compute nodes and the head node on a shared network disk.
- Make sure that the license manager is available and up and running.
- If you work on a desktop PC, which is recommended, install COMSOL on that local PC. Also install Windows HPC Pack on the desktop PC before you start. Windows HPC Pack makes it possible to access the cluster from your workstations. It is free and ships with the Microsoft® HPC Server 2008 (HPCS 2008). An alternative is to log in to the cluster via Remote Desktop, for example.

To run a cluster job, follow these steps:

1 Start COMSOL Multiphysics.

2 In a complete model, right-click the **Study** node and select **Cluster Computing** ( ).

3 You can make a distributed parametric sweep to take advantage of the cluster with any solver settings. If you want to use a distributed solver job for other models without a parametric sweep, make sure that you enable a direct solver for the active solver node under **Solver Configurations**. To do so, right-click the **Direct** node ( ) and select **Enable**. The MUMPS and SPOOLES direct solvers support distributed solver jobs.

If the **Solver Configurations** node does not exist, right-click the **Study** node and select **Show Default Solver** ( ).

4 In the **Cluster Computing** node's settings window, select **HPCS 2008** from the **Cluster type** list. This provides access to all parameters that you need for communication with the cluster.

5 To submit the job, right-click the **Study** node ( ) and select **Compute** (=).

6 You can define more details in the settings window for the **Cluster Computing** node ( ).under **Job Configurations** ( ). When you submit a job, COMSOL Multiphysics adds a **Cluster Computing** node. If you want to change or inspect its settings before submitting the first job, right-click the **Study** node ( ) and select **Show Default Solver** ( ).

7 After submitting the job to the cluster, you can monitor the progress in the **Progress** window and the **Log** window. The **Progress** window shows the progress of the batch data and external processes, and the **Log** window contains a log with information about the solver operations for each parameter in a parametric sweep, for example. You can also get details about a cluster job in the Windows Job Manager, which is available in the HPC Pack.

RUNNING A CLUSTER JOB ON LINUX

Before you begin, make sure that the license manager is up and running and reachable from all compute nodes and the headnode. Skip the steps 1) and 3) if you are running COMSOL on the machine from where you wish to start the cluster job.

1 Start the COMSOL server on the Linux® system with the command `comsol server`. Notice the port number that is displayed (for example, *COMSOL 4.4 started listening on port 2036*).

2 Start COMSOL Multiphysics on your desktop computer.

3 From the **File** menu, choose **Client Server>Connect to Server** ( ). In the **Connect to Server** dialog box, use the login credentials that you entered at the startup of the COMSOL server.

4 In a complete model, right-click the **Study** node and select **Cluster Computing** ( ).

5 You can make a distributed parametric sweep to take advantage of the cluster with any solver settings. If you want to use a distributed solver job for other models without a parametric sweep, make sure that you enable a direct solver for the active solver node under **Solver Configurations**. To do so, right-click the **Direct** node ( ) and select **Enable**. The MUMPS and SPOOLES direct solvers support distributed solver jobs.

If the **Solver Configurations** node does not exist, right-click the **Study** node and select **Show Default Solver** ( ).

6 In the **Cluster Computing** node's settings window, select **General** from the **Cluster type** list for Linux clusters. This provides access to all parameters that you need for communication with the cluster.

7 To submit the job, right-click the **Study** node ( ) and select **Compute** (=).

8 You can define more details in the settings window for the **Cluster Computing** node ( ) under **Job Configurations** ( ). When you submit a job, COMSOL Multiphysics adds a **Cluster Computing** node. If you want to change or inspect its settings before submitting the first job, right-click the **Study** node ( ) and select **Show Default Solver** ( ).

9 After submitting the job to the cluster, you can monitor the progress in the **Progress** window and the **Log** window. The **Progress** window shows the progress of the batch data and external processes, and the **Log** window contains a log with information about the solver operations for each parameter in a parametric sweep, for example.

You can do the same cluster simulation from the command line using, for example, a scheduler script. Some example commands:

```
comsol -nn 2 mpd boot -f machinefile -v -d
comsol -nn 2 batch -inputfile comsoltest.mph -outfile output.mph -batchlog b.log
comsol mpd allexit
```

CLUSTER LICENSE HANDLING

To run COMSOL Multiphysics simulations in distributed-memory parallel mode (on a cluster), you must have a floating network license (FNL). Look for the keyword CLUSTERNODE in your license file. When running a cluster job, COMSOL uses the following license components and license check-out procedures:

- On the headnode, one seat of the COMSOL and COMSOLGUI features each are checked out.
- On each of the compute nodes, only one CLUSTERNODE feature is checked out, and it is not counted. This means that you have unlimited number of cluster nodes available for every seat (job) of the floating network license.
- When running a batch job through a scheduler, COMSOL's license manager checks out the noncluster COMSOL license keys (COMSOL, COMSOLGUI, CADIMPORT, CHEM, and so on) from one of the distributed processes. All other processes in the batch job only check out a CLUSTERNODE license key. So, license keys can be checked out from any physical node in the cluster depending on where the scheduler starts the processes.

The COMSOL Commands

The following sections describe the `comsol` commands on the Windows®, Linux®, and Macintosh platforms.

COMSOL Commands on Windows

Use a COMSOL command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL commands is

```
<command> [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. There are several different commands (See `<command>` in the command syntax) that can be combined with optional targets to achieve various results. The table below lists the major available commands and targets (if the Availability column is empty, the command is always available):

TABLE 21-1: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
`comsol`	Run standalone COMSOL Desktop	
`comsolxpl`	Run cross-platform COMSOL Desktop	
`comsolserver`	Start COMSOL server	
`comsol client`	Run COMSOL Desktop client	
`comsolbatch`	Run a COMSOL MPH-file or class file	
`comsolcompile`	Compile a model file for Java	
`comsolcluster`	Run COMSOL Desktop on a cluster	Requires a floating network license (FNL)
`comsolclusterxpl`	Run cross-platform COMSOL Desktop on a cluster	Requires a floating network license (FNL)
`comsolclusterbatch`	Run COMSOL cluster version in batch mode	Requires a floating network license (FNL)
`comsolclusterserver`	Run COMSOL cluster server	Requires a floating network license (FNL)
`comsolserver matlab`	Start MATLAB® and connect to a COMSOL server	Requires a LiveLink™ *for* MATLAB® license
`comsol convertpre35a`	Convert 3.0–3.5 models	

The commands are available in 32-bit versions in the subdirectory `bin\win32` in the COMSOL installation directory, and in 64-bit versions in the subdirectory `bin\win64` in the COMSOL installation directory. The COMSOL installer sets up a few of the possible commands on your start menu and your desktop. In Windows 8, you can click the shortcut **COMSOL Launchers** on the Apps screen. This makes a folder with shortcuts to all COMSOL commands available.

To create additional customized commands, you can create shortcuts including all argument and put them on your desktop. You can also issue COMSOL commands in a command window. To conveniently access the command in a command window, you need to set up the Windows path to include one the paths `bin\win32` or `bin\win64` in the COMSOL installation directory.

INI FILES

For each launcher file, there is a corresponding `.ini` file in the same directory. It is sometimes recommended that these files are edited. For example, you can add options to any of the above commands by modifying the corresponding INI file.To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsolbatch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the COMSOL command and target. Table 21-2 lists the options (See [`<options>`] in the command syntax) available for all COMSOL commands. Always issue these options between the command and the target (if any).

TABLE 21-2: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-h`	Print general help	
`<target> -h`	Print target-specific help	
`-3drend ogl | dx9 | sw`	3D renderer: OpenGL, DirectX, or software rendering	
`-docroot <path>`	Specify custom path to the COMSOL documentation root directory.	See Documentation and Model Libraries Root Directories
`-modelsroot <path>`	Specify custom path to the COMSOL Model Libraries root directory.	See Documentation and Model Libraries Root Directories
`-np <no. of processors>`	Number of processors	See Shared-Memory Options
`-numasets <no. of sets>`	Number of NUMA sets	See Shared-Memory Options
`-numafirst <numa number>`	Set first NUMA node to bind process to	See Shared-Memory Options
`-mpmode throughput | turnaround | owner`	Multiprocessor mode	See Shared-Memory Options
`-blas {auto} | mkl | acml | path`	BLAS library to use	See BLAS Options
`-blaspath <path>`	BLAS library path	See BLAS Options
`-ipv6`	Activate IPv6 support	
`-c <path>`	License file path	
`-prefsdir <path>`	Preference directory	
`-tmpdir <path>`	Temporary file directory	
`-version`	Print COMSOL version	
`-version <target>`	Print target version	
`-ckl`	Use class-kit license	

TABLE 21-2: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-autosave {on}|off`	Control saving of recovery files	
`-recoverydir <path>`	Path to recovery directories	

For the `-tmpdir` option, COMSOL uses the specified directory to store temporary files. Use the `-prefsdir` option to specify the directory where COMSOL stores the preference file.

Documentation and Model Libraries Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Libraries root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Libraries root directories can be useful for administering Model Library Update; see The Model Library Update Window.

> Setting the paths to the documentation and Model Libraries root directories using these options does not in itself move the directories and their contents.

Shared-Memory Options

- Use the option `-np` to control the number of core and processors used. The default is to use all available cores and processors.
- Use the option `-numasets` to control the number of Non-Uniform Memory Access node sets COMSOL should take into account. This is usually the number of processor sockets that the hardware is using.
- Depending on how loaded the machine is, you can control how COMSOL uses the available processors with the `-mpmode` option. The following options are available:

TABLE 21-3: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
`throughput`	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
`turnaround`	Typically provides the best performance when no other processes than COMSOL are active.
`owner`	Provides the highest performance in most cases.

Sometimes you might want to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides for the following BLAS-related options:

TABLE 21-4: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL for Intel processors, and ACML for AMD processors with SSE2 support, otherwise MKL. (This is the default option.)
mkl	Use the Intel MKL library.
acml	Use the AMD ACML library.
path	Use a BLAS library specified using the option `-blaspath` or the environment variable `COMSOL_BLAS_PATH`.

Both MKL and ACML are distributed along with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in `/lib/ARCH`, where `ARCH` is the architecture (`win32` or `win64`) or somewhere in the standard search path. Also provide the path to any sublibraries needed by the library. Set the search path to point to the directory where the library is installed. To do so, use the environment variable `PATH`. Your library must support both the standard FORTRAN LAPACK interface and the standard FORTRAN BLAS interface. If your LAPACK and BLAS interface consists of several libraries, use the path to the LAPACK library.

COMSOL COMMANDS

In additions to the options in Table 21-2, the standalone COMSOL command supports the following option:

TABLE 21-5: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
`-open <file>`	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ *for* MATLAB® also needs to connect to a COMSOL server.

The Windows syntax for the COMSOL server command is

```
comsolserver [<options>] [<target arguments>]
```

The following target arguments are available for a COMSOL server command:

TABLE 21-6: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTION	DESCRIPTION
`-user <user>`	Specify login name for a user.
`-port <port>`	Specify a TCP/IP port to listen for connect attempts.
`-passwd reset|nostore`	Specify that you want to provide a new password.To avoid storing the new password on file use nostore.
`-login {info}|force|never|auto`	Ask for login information. `info` means that only missing information is asked for. `force` resets the password. `never` requires that the login information is available. `auto` automatically creates a new username and password.
`-multi on|{off}`	Accept repeated client connections.
`-silent`	Do not listen to standard input.
`-graphics`	Start the server with graphics libraries. This displays plots on the server when you are connected with a client (that is, not with the COMSOL GUI).

Accessing the COMSOL Server Computer

The server computer can be accessed in several ways. If it is dedicated to a single person, you can sit down at that machine and log in on it. You can also connect to the server computer by using Remote Desktop. Start the COMSOL server from the **Start** menu. If several people want to access a single Windows computer to run the COMSOL server, you must use Windows Terminal Server or another tool that allows multiple users to log in on the same Windows server. In some Windows versions, Microsoft® provides a Telnet Server with which you can log in through a terminal window. When using a terminal window to log in on Windows, use the `comsolserver` command to start the COMSOL server.

Login Information

When a COMSOL server is started for the first time, you are asked for a username and password. Select a username and a password, which COMSOL Multiphysics then uses in communications between the COMSOL client and the server. You must also specify a matching username and password in the **Connect to Server** dialog box. The software writes this login information in the subdirectory `.comsol/v44/login.properties` in your Windows home directory.

Client/Server Security Issues

COMSOL Multiphysics can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.

! Always make sure that untrusted users cannot access the COMSOL login information. Protect the file `.comsol/v44/login.properties` in your home directory. This is important when running COMSOL in client/server mode. Alternatively, start the COMSOL server with the `-passwd nostore` option, and clear **Remember Password** when connecting to the server. This ensures that your login information is not stored on file.

Once a COMSOL server is started, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the COMSOL server, both in your firewall and by changing the COMSOL server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>\bin\conf\server.xml` and find the lines:

```
<!-- To restrict access to the COMSOL server you can uncomment the block below.
```

and follow the instructions. The default port for the COMSOL server is 2036. You can change this by using the option `-port <port>` when launching COMSOL and COMSOL server.

Documentation Security Issues

To serve the COMSOL Desktop with documentation COMSOL opens a separate documentation server on the client computer when you open the documentation.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the documentation server, both in your firewall and by changing the documentation server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>\doc\help\conf\server.xml` and find the lines:

```
<!-- To restrict access to the documentation server you can uncomment the block below.
```

and follow the instructions. The default port for the documentation server is 8090. You can change this by using the option `-docport <docport>` when launching COMSOL.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with a the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command:

TABLE 21-7: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
`-port <port>`	Specify a TCP/IP port to connect to
`-server <server name>`	Specify server to connect to
`-open <file>`	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. Run both Model MPH-files and model files for Java with the COMSOL batch command. Model files for Java need to be compiled before running.

The Windows syntax for the COMSOL batch command is

```
comsolbatch [<options>] [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-8: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH TARGET ARGUMENTS	DESCRIPTION
`-inputfile <file name>`	Run a Model MPH-file or class file.
`-outputfile <file name>`	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output.
`-job <job tag>`	The batch job to run.
`-study <study tag>`	The study to compute.
`-pname <parameter name>`	Comma-separated list of parameter names.
`-plist <parameter value>`	Comma-separated list of parameter values.
`-batchlog <filename>`	File to store log in.
`-client`	Run as client.
`-host`	Connect to host.
`-port`	Connect to port.
`-graphics`	Start COMSOL batch with graphics libraries. This displays plots during analysis.
`-checklicense <filename>`	Print license requirements for a Model MPH-file.
`-nosave`	Do not save the resulting model.

Example

To use the COMSOL Batch mode to solve a model, run the following command:

```
comsolbatch -inputfile in.mph -outputfile out.mph -study std1
```

This command starts COMSOL Batch, solves the model in the Model MPH-file with the given filename (`in.mph` in this example) using the active solver settings in the model, and stores the solution in the `out.mph`.

The `-study` option directs COMSOL to run a certain study. The study is identified by its tag. In the COMSOL Desktop, select **Show Name and Tag** under **Model Builder Node Label** to see the tags of the jobs under **Study** within curly braces in the Model Builder. In the model object, determine the tags of the jobs by the command `model.study().tags()`. You can determine the name of each study by `model.study(<tag>).name()` using one of the job tags.

The `-job` option works similar to the `-study` option. It directs COMSOL to start a certain job. The job is identified by its tag. In the model object, determine the tags of the jobs by the command `model.batch().tags()`. You can determine the name of each job by `model.batch(<tag>).name()` using one of the job tags.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a model file for Java for use by the COMSOL batch command or for loading class files into the GUI. The Windows syntax for the COMSOL compile command is

```
comsolcompile [<options>] [<target arguments>] <file>.java
```

The Java file is mandatory. The following optional target arguments are available:

TABLE 21-9: COMSOL CLUSTER TARGET ARGUMENTS

COMSOL COMPILE TARGET ARGUMENTS	DESCRIPTION
`-jdkroot <path>`	Path to the JDK root
`-classpathadd <classpath>`	Additional classpath
`-verbose`	Verbose output

COMSOL CLUSTER COMMANDS

All COMSOL cluster commands require a floating network license.

To start a COMSOL Desktop running in distributed mode on a Windows cluster, type

```
mpiexec -n <number of nodes> comsolcluster.exe <options> [<target arguments>]
```

To start a COMSOL server running in distributed mode on a Windows cluster, type

```
mpiexec -n 1 comsolserver.exe [<options>] <target arguments>] -cluster on : -n <number of nodes-1>
comsolclusterserver.exe <options> [<target arguments>]
```

Note that all options and target arguments need to be repeated twice, when using the two above commands.

To start a cross-platform COMSOL Desktop running in distributed mode on a Windows cluster, type

```
mpiexec -n <number of nodes> comsolclusterxpl.exe <options> [<target arguments>]
```

To start a COMSOL batch command running in distributed mode on a Windows cluster, type

```
mpiexec -n <number of nodes> comsolclusterbatch.exe <options> [<target arguments>]
```

The following cluster commands are available:

TABLE 21-10: COMSOL CLUSTER TARGETS

COMSOL CLUSTER COMMANDS	DESCRIPTION
`comsolclusterbatch`	Run COMSOL on a cluster in batch mode
`comsolclusterserver`	Run COMSOL server on a cluster
`comsolcluster`	Run COMSOL Desktop on a cluster
`comsolclusterxpl`	Run the cross-platform COMSOL Desktop on a cluster

The preferred way of starting COMSOL jobs is from the Job Configurations node in the COMSOL Desktop's model tree.

Micromixer—Cluster Version: model library path
COMSOL_Multiphysics/Tutorial_Models/micromixer_cluster

If you need to start COMSOL cluster jobs from the command line, the preferred way is to use the `comsolclusterbatch` command because the `comsolclusterserver` and `comsolcluster` commands require TCP/IP access from your client computer to the cluster node where COMSOL runs.

The Windows Configuration

- Make sure that Windows HPC Server 2008 or Windows Compute Cluster Server 2003 is installed. Running distributed COMSOL on other Windows versions is not supported.
- Make sure that the Windows HPC Server 2008 working directory is set to point to the `comsol` command directory (`<path to COMSOL install directory>`
  `\bin\win64`). The install directory must be shared between the nodes on your cluster. In some network configurations the firewall prevents you from starting MPI on a shared executable. To register the executable with the firewall use the `clusrun` command to execute the `hpcfwutil` command on all nodes, for instance to register `comsolclusterbatch` use `clusrun /all hpcfwutil register comsolclusterbatch.exe <shared path to COMSOL install directory>\bin\win64\comsolclusterbatch.exe`
- Also make sure that the Microsoft Visual Studio 2010 and 2008 Runtimes are installed on all nodes. They are called `vcredist_*.exe`. You can install them from the root directory of the DVD using the `clusrun` command, for instance.
- Also make sure that all nodes that you intend to run COMSOL on have access to the license manager and that you can start COMSOL running in nondistributed mode. The nodes require access to the license manager to check out licenses.

Example of the COMSOL Batch Command

Schedule a job with the command

```
mpiexec -n -1 comsolclusterbatch.exe -np 2 -inputfile <filename>
```

to run a COMSOL batch on a number of computational nodes given by `mpiexec`. For further information about the `mpiexec` command and Windows HPC Server 2008, consult the documentation that was shipped with the product and the online manuals.

Example of the COMSOL Server Command

When a COMSOL server cluster job is created, a preference directory must be set and be reachable from all nodes to avoid problems with the server login; see The COMSOL Commands and Login Information. The preferences can be generated by starting COMSOL server once on the head node using the command

```
comsolserver.exe -prefsdir <prefsdir>
```

where `<prefsdir>` is a preference directory *common* to all nodes.

When the COMSOL server is started on the cluster, the port number is written to standard output, so a standard output file and a standard error file must be set for the cluster job. To start a COMSOL server, schedule a job with the following command:

```
mpiexec -n 1 comsolserver.exe -np 2 -prefsdir <prefsdir> -cluster on : -n -1
comsolclusterserver.exe -np 2 -prefsdir <prefsdir>
```

The argument `-1` indicates that the number of computational nodes is decided at the `mpiexec` launch. You must be able to access the cluster node where the COMSOL server runs from the COMSOL client computer.

COMSOL MPI Options

The COMSOL cluster target arguments specify what MPI library to use and what Scalapack version to use. There are several implementations of MPI. COMSOL by default uses the Windows HPC Server 2008 or Windows CCS 2003 MPI libraries. COMSOL also supports most MPI implementations based on MPICH2. It is recommended that the default library is used. COMSOL also has a compatibility mode, which you activate by adding the option `-mpi mpich2`. When using this option both the variables `PATH` and `LD_LIBRARY_PATH` must include the MPI implementation. It is also possible to use other MPI libraries based on MPICH2 using the option `-mpipath <path to shared library>`. The following target arguments are available for a COMSOL cluster commands:

TABLE 21-11: COMSOL CLUSTER TARGET ARGUMENTS

COMSOL BATCH TARGET ARGUMENTS	DESCRIPTION
`-mpi {auto} | mpich2 | wccs2003 | whpc2008 | user | path`	MPI library to use
`-mpipath <path>`	MPI library path
`-scalapack {auto} | mpich2 | wccs2003 | whpc2008 | user | path`	Scalapack library to use
`-scalapackpath <path>`	Scalapack library path

The Cluster Computing study allows you to set up a batch job for submission to a Windows HPC Server 2008 job scheduler or Windows Compute Cluster Server 2003 job scheduler. There are several settings that you can configure in the `comsol.ini` file to get default settings:

```
-Dcs.scheduler=<IP or network adress>
-Dcs.clusteruser=<Username on cluster>
-Dcs.rundir=<Where the model file is located on the cluster>
-Dcs.comsoldir=<Installation path to comsol on the cluster>
```

Additionally you can configure the commands:

```
-Dcs.precmd=<Command line>
-Dcs.postcmd=<Command line>
```

This adds commands prior to the `comsol` command and after the `comsol` command. You can add `{nn}` or `{perhost}` to any of these pre- or postcommands. This configures the Cluster Computing study to use the number of nodes and number of nodes on each host from the corresponding settings for the Cluster Computing study. For more information, see Cluster Computing.

COMSOL MATLAB COMMAND

Use the COMSOL `matlab` command to access the COMSOL API through MATLAB. Enter the following command:

```
comsolserver matlab
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsolserver matlab` command:

TABLE 21-12: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
`-mlroot <path>`	MATLAB installation directory.
`-host <hostname>`	Connect to host.
`-port <hostname>`	Connect to port.
`-desktop`	Start with Desktop.
`-nodesktop`	Start without Desktop.
`-mlnosplash`	Start without MATLAB splash screen.
`-graphics`	Start the server with graphics libraries. This enables plotting on the server. Available only when running `comsolserver matlab [<options>]`.

COMSOL Commands on Linux

Use the `comsol` command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL command is

```
comsol [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. The `comsol` command can be combined with optional targets to achieve various results. The table below lists the command and targets:

TABLE 21-13: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
`comsol`	Run standalone COMSOL Multiphysics	
`comsol server`	Start COMSOL Multiphysics server	
`comsol client`	Run COMSOL Multiphysics client	
`comsol batch`	Run a COMSOL MPH-file or class file	
`comsol compile`	Compile a model file for Java	
`comsol server matlab`	Start MATLAB® and connect to a COMSOL server	Requires LiveLink™ *for* MATLAB® license
`comsol convertpre35a`	Convert 3.0–3.5 models	
`comsol mpd`	Run the COMSOL multiprocessing daemon	Requires CLUSTERNODE license

The `comsol` command is located in the `bin` folder in the COMSOL installation directory.

INI FILES

There is a number of `.ini` files in the subdirectories `glnx86` and `glnxa64` in the `bin` directory. It is sometimes recommended that you edit these files. For example, you can add options to any of the above commands by modifying the corresponding `ini` file. To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsol batch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the `comsol` command and target. Table 21-14 lists the options (See `[<options>]` in the command syntax) available for all comsol commands. Always issue these options between the command and the target (if any).

TABLE 21-14: COMSOL OPTIONS (CURLY BRACES INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-h`	Print general help	
`<target> -h`	Print target-specific help	
`-32`	Run 32-bit COMSOL	
`-64`	Run 64-bit COMSOL	

TABLE 21-14: COMSOL OPTIONS (CURLY BRACES INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-3drend ogl	sw`	3D renderer: OpenGL or software rendering	
`-comsolinifile`	Specify custom path to .ini-file used when starting COMSOL	
`-docroot <path>`	Specify custom path to the COMSOL documentation root directory	See Documentation and Model Libraries Root Directories
`-modelsroot <path>`	Specify custom path to the COMSOL Model Libraries root directory	See Documentation and Model Libraries Root Directories
`-np <no. of processors>`	Number of processors	See Shared-Memory Options
`-numasets <no. of sets>`	Number of NUMA sets	See Shared-Memory Options
`-numafirst <numa number>`	Set first NUMA node to bind process to	See Shared-Memory Options
`-mpmode throughput	turnaround	owner`	Multiprocessor mode	See Shared-Memory Options
`-blas {auto}	mkl	acml	path`	BLAS library to use	See BLAS Options
`-blaspath <path>`	BLAS library path	See BLAS Options
`-ipv6`	Activate IPv6 support	
`-nn <no. of nodes>`	Number of nodes	See COMSOL Cluster Commands
`-nnhost <no. of nodes>`	Number of nodes on each host	See COMSOL Cluster Commands
`-f <path>`	Path to hostfile	See COMSOL Cluster Commands
`-mpi {auto}	intel	mpich2	wccs2003	whpc2008	user	path`	MPI library to use	See COMSOL Cluster Commands
`-mpipath <path>`	MPI library path	See COMSOL Cluster Commands
`-mpiroot <path>`	MPI library root path	See COMSOL Cluster Commands
`-mpirsh {rsh}	ssh`	Use rsh or ssh when booting MPD	See COMSOL Cluster Commands
`-mpibootstrap {ssh}	rsh	fork	slurm	ll	lsf	sge	jmi`	Set bootstrap server for Hydra	See COMSOL Cluster Commands
`-mpibootstrapexec <path>`	Executable used by bootstrap server	See COMSOL Cluster Commands
`-mpidebug <debug level>`	Set the MPI output level	See COMSOL Cluster Commands
`-mpienablex`	Enable Xlib forwarding	See COMSOL Cluster Commands

TABLE 21-14: COMSOL OPTIONS (CURLY BRACES INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-mpifabrics fabric1:fabric2`	Select network fabrics where fabric1 is one of <shm | dapl | tcp | tmi | ofa>, and fabric2 is one of <dapl | tcp | tmi | ofa>	See COMSOL Cluster Commands
`-mpd`	Use MPD instead of Hydra launcher	See COMSOL Cluster Commands
`-scalapack {auto}` | `mpich2` | `wccs2003` | `whpc2008` | `user` | `path`	Scalapack library to use	See COMSOL Cluster Commands
`-scalapackpath <path>`	Scalapack library path	See COMSOL Cluster Commands
`-clustersimple`	Simple startup of cluster	See COMSOL Cluster Commands
`-c <path>`	License file path	
`-prefsdir <path>`	Preference directory	
`-tmpdir <path>`	Temporary file directory	
`-version`	Print COMSOL version	
`-version <target>`	Print target version	
`-ckl`	Use class-kit license	
`-forcegcc`	Force load of GCC libraries	
`-forcecomsolgcc`	Force load of GCC libraries shipped with COMSOL	
`-autosave <{on}` | `off>`	Control saving of recovery files	
`-recoverydir <path>`	Path to recovery directories	

On 32 bit systems, the option `-32`, is the default option, and the 64 bit option is not available. On 64 bit systems, the option `-64`, is the default option, and the 32 bit option may work if you have installed required 32 bit runtime libraries. These are installed by default on some 64 bit distributions.

For the `-tmpdir` option, COMSOL uses the specified directory to store temporary files. Use the `-prefsdir` option to specify the directory where COMSOL stores the preference file.

Documentation and Model Libraries Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Libraries root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Libraries root directories can be useful for administering Model Library Update; see The Model Library Update Window.

Setting the paths to the documentation and Model Libraries root directories using these options does not in itself move the directories and their contents.

Shared-Memory Options

Use the option `-np` to control the number of cores and processors used. The default is to use all available cores and processors.

Use the option `-numasets` to control the number of non-uniform memory access (NUMA) node sets COMSOL should take into account. This is usually the number of processor sockets that the hardware is using.

Depending on how loaded your machine is, you can control how COMSOL uses the available processors. The following options are available:

TABLE 21-15: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
`throughput`	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
`turnaround`	Typically provides the best performance when no other processes than COMSOL are active.
`owner`	Provides the highest performance in most cases.

Sometimes you might want to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides for the following BLAS related options:

TABLE 21-16: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL for Intel processors, and ACML for AMD processors with SSE2 support, otherwise MKL. (This is the default option.)
mkl	Use the Intel MKL library.
acml	Use the AMD ACML library.
path	Use a BLAS library specified using the option `-blaspath` or the environment variable `COMSOL_BLAS_PATH`.

Both MKL and ACML are distributed with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in `/lib/ARCH`, where `ARCH` is the architecture (`glnx86` or `glnxa64`), or somewhere in the standard search path. You must also provide the path to any sublibraries needed by the library. You can also set the search path to point to the directory where the library is installed. To do so, use the environment variable `LD_LIBRARY_PATH`. Your library must support both the standard FORTRAN LAPACK interface and the standard FORTRAN BLAS interface. If your LAPACK and BLAS interface consists of several libraries, use the path to the LAPACK library.

GCC Options

By default COMSOL uses the GCC libraries installed on the system. If COMSOL is unable to start, the software uses the GCC libraries shipped with COMSOL. To force COMSOL to use the shipped GCC libraries, use the `-forcecomsolgcc` option. The option `-forcegcc` is mainly intended for use together with the LiveLink™ *for* MATLAB®; use it if you are unable to make function call backs to MATLAB.

COMSOL COMMANDS

In additions to the options in Table 21-14, the standalone COMSOL command supports the following option.

TABLE 21-17: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
`-open <file>`	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ *for* MATLAB® also needs to connect to a COMSOL server.

The syntax for the COMSOL server command is

```
comsol [<options>] server [<target arguments>]
```

The following target arguments are available for a COMSOL server command.

TABLE 21-18: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTIONS	DESCRIPTION
`-user <user>`	Specify login name for a user.
`-port <port>`	Specify a TCP/IP port to listen for connect attempts.
`-passwd reset|nostore`	Specify that you want to provide a new password.To avoid storing the new password on file use <nostore>.
`-login {info}|force|never|auto`	Ask for login information. `info` means that only missing information is asked for. `force` resets the password. `never` requires that the login information is available. `auto` automatically creates a new username and password.
`-multi on|{off}`	Accept repeated client connections.
`-silent`	Do not listen to standard input.
`-graphics`	Start the server with graphics libraries. This displays plots on the server when you are connected with a client (that is, not with the COMSOL GUI).

Accessing the COMSOL Server Computer

To access the computer running the COMSOL server simply log in on the server computer by using `ssh` or a similar command, then enter the `comsol server` command.

Login Information

When you start a COMSOL server for the first time, you are asked for a username and password. Select a username and a password, which COMSOL then uses in communications between the COMSOL client and the server. You must also specify a matching username and password in the **Connect to Server** dialog box. The software writes this login information in the subdirectory `.comsol/v44/login.properties` in your home directory.

Client/Server Security Issues

COMSOL Multiphysics can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.

! Always make sure that untrusted users cannot access the COMSOL login information. Protect the file `.comsol/v44/login.properties` in your home directory. This is important when using the COMSOL client/server configuration. Alternatively, start the COMSOL server with the `-passwd nostore` option, and clear **Remember Password** when connecting to the server. This ensures that your login information is not stored on file.

Once you start a COMSOL server, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the COMSOL server, both in your firewall and by changing the COMSOL server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>/bin/conf/server.xml` and find the lines:

```
<!-- To restrict access to the COMSOL server you can uncomment the block below.
```

and follow the instructions. The default port for the COMSOL server is 2036. You can change this by using the option `-port <port>` when launching COMSOL and COMSOL server.

Documentation Security Issues

To serve the COMSOL Desktop with documentation COMSOL opens a separate documentation server on the client computer when you open the documentation.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the documentation server, both in your firewall and by changing the documentation server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>/doc/help/conf/server.xml` and find the lines:

```
<!-- To restrict access to the documentation server you can uncomment the block below.
```

and follow the instructions. The default port for the documentation server is 8090. You can change this by using the option `-docport <docport>` when launching COMSOL.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with a the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command:

TABLE 21-19: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
`-port <port>`	Specify a TCP/IP port to connect to
`-server <server name>`	Specify server to connect to
`-open <file>`	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. You can run both Model MPH files and model files for Java with the COMSOL batch command. Model files for Java need to be compiled before running.

The syntax for the COMSOL batch command is

```
comsol [<options>] batch [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-20: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
`-inputfile <file name>`	Run a Model MPH-file or class file.
`-outputfile <file name>`	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output.
`-job <job tag>`	The batch job to run.
`-study <study tag>`	The study to compute.
`-pname <parameter name>`	Comma separated list of parameter names.
`-plist <parameter value>`	Comma separated list of parameter values.
`-batchlog <file name>`	File to store log in.
`-client`	Run as client.
`-host`	Connect to host.
`-port`	Connect to port.
`-graphics`	Start COMSOL batch with graphics libraries. This displays plots during postprocessing.
`-checklicense <filename>`	Print license requirements for a Model MPH-file.
`-nosave`	Do not save the resulting model.

Example

To use the COMSOL batch mode to solve a model, run the following command:

```
comsol batch -inputfile in.mph -outputfile out.mph -study std1
```

This command starts COMSOL in batch mode, solves the model in the Model MPH-file with the given file name using the active solver settings in the model, and stores the solution in the `out.mph`.

The `-study` option directs COMSOL to run a certain study. The study is identified by its tag. In the COMSOL Desktop, select **Show Name and Tag** under **Model Builder Node Label** to see the tags of the jobs under **Study** within curly braces in the Model Builder. In the model object, determine the tags of the jobs by the command `model.study().tags()`. You can determine the name of each study by `model.study(<tag>).name()` using one of the job tags.

The `-job` option works similar to the `-study` option. It directs COMSOL to start a certain job. The job is identified by its tag. In the model object, determine the tags of the jobs by the command `model.batch().tags()`. You can determine the name of each job by `model.batch(<tag>).name()` using one of the job tags.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a model file for Java for use by the COMSOL batch command or for loading class files into the GUI. The syntax for the COMSOL compile command is

```
comsol [<options>] compile [<target arguments>] <file>.java
```

The Java file is mandatory. The following optional target arguments are available:

TABLE 21-21: COMSOL COMPILE OPTIONS

COMSOL COMPILE OPTIONS	DESCRIPTION
`-jdkroot <path>`	Path to the JDK root

TABLE 21-21: COMSOL COMPILE OPTIONS

COMSOL COMPILE OPTIONS	DESCRIPTION
`-classpathadd <classpath>`	Additional classpath
`-verbose`	Verbose output

COMSOL CLUSTER COMMANDS

Use the COMSOL command with the option `-nn <no. of nodes>` to run COMSOL on clusters.

The syntax for the COMSOL cluster command is

```
comsol -nn <no. of nodes> [<options>] [<target>] [<target arguments>]
```

The following cluster commands are available:

TABLE 21-22: COMSOL CLUSTER TARGETS

COMSOL CLUSTER COMMANDS	DESCRIPTION
`comsol -nn <nn> batch`	Run COMSOL on a cluster in batch mode
`comsol -nn <nn> server`	Run COMSOL server on a cluster
`comsol -nn <nn>`	Run COMSOL Desktop on a cluster

The preferred way of starting COMSOL cluster jobs is from the Study node in the COMSOL Desktop. If you need to start COMSOL cluster jobs from the command line, the preferred way is to use the `comsol -nn <nn> batch` command because the `comsol -nn <nn> server` and `comsol -nn <nn>` commands require TCP/IP access from your client computer to the cluster node where COMSOL runs.

Running on Linux

COMSOL uses Hydra by default to initialize the MPI environment. Hydra is more scalable than MPD and it does not require any additional commands to launch.

Troubleshooting Distributed COMSOL and MPI

To launch COMSOL with Hydra use the command line

```
comsol -nn <number of compute nodes> -f <filename>
```

The file `<filename>` should contain the hostnames of the compute nodes that you intend to use. You can find out the hostname of each node from the Linux command `hostname`. Each node should be listed on a separate line in the file. You can also list the IP address of each node. The file can contain more compute nodes than you actually intend to use.

- You can set the remote node access mechanism that is used for connecting using the switch `-mpibootstrap`. The valid options are `ssh`, `rsh`, `fork`, `slurm`, `ll`, `lsf`, `sge`, and `jmi`. This is important if the cluster only supports a different remote node access mechanism than `ssh` because `ssh` is the default protocol used.
- Use the switch `-mpibootstrapexec` to set the path to the remote node access mechanism such as `/usr/bin/ssh`.
- The option -mpidebug sets the output level from MPI. The default is level 4.
- You can control the network fabrics used for communication with the option `-mpifabrics fabric1:fabric2` where `fabric1` is one of `shm`, `dapl`, `tcp`, `tmi`, or `ofa`, and `fabric2` is one of `dapl`, `tcp`, `tmi`, or `ofa`. Use this option if you are having trouble with the default fabrics used.
- Use `-mpienablex` to enable Xlib forwarding. Xlib forwarding is off by default.

Previously there was a shorthand for performing the COMSOL MPI environment initialization and starting COMSOL. The `-clustersimple` option is still supported but is equivalent to the Hydra command by default, for example

```
comsol -nn 4 -clustersimple
```

You should usually run COMSOL in batch mode. Use the command

```
comsol -clustersimple batch -inputfile input.mph -outputfile output.mph
```

It allows the Intel MPI library to automatically detect the number of nodes that were scheduled to the program. Restricting the number of processes with the `-nn` switch allows COMSOL to combine MPI with multithreading. This is the most efficient way to run COMSOL.

Using the MPD launcher

In previous versions of COMSOL, the MPI environment was launched by MPD. You can still use MPD if you use the switch `-mpd`, but it is recommended that you use Hydra.

In order to start MPI, have a file named `.mpd.conf` in your home directory to which you alone have access. This file should contain the single line

```
secretword = <your secret word here>
```

On Intel MPI shipped with COMSOL, the `.mpd.conf` file is optional.

Below, the details of the individual cluster commands are described.

Before you start COMSOL, you must initialize the MPI environment. A so-called multiprocessing daemon (MPD) must run on each computer node that you intend to use. To start MPD on several computer nodes, enter

```
comsol -nn <number of compute nodes> mpd boot -f <filename>
```

The file `<filename>` should contain the hostnames of the compute nodes that you intend to use. You can find out the hostname of each node from the Linux command `hostname`. Each node should be listed on a separate line in the file. You can also list the IP address of each node. The file can contain more compute nodes than you actually intend to use. As an alternative to using the `-f <filename>` option, you can put the list of compute nodes in a file named `mpd.hosts` in your home directory. You can set the protocol that is used for connecting using the switch `-mpirsh`. The valid options are `rsh` and `ssh`. This is important if the cluster only supports `ssh` because `rsh` is the default protocol used. Make sure that all nodes were booted by listing them with the command

```
comsol mpd trace
```

Start distributed COMSOL with the `-nn` option. For example, enter

```
comsol -nn <number of computational nodes> -mpd server
```

to start a COMSOL server running on a specific number of computational nodes. The number of computational nodes can exceed the number of compute nodes. Use the `-nnhost` option if you want to force several computational nodes on a compute node. Avoid starting more computational nodes than the total number of cores that you have available on a compute node. When you have finished using distributed COMSOL, you should take down all the MPDs. Enter the command

```
comsol mpd allexit
```

to stop all MPDs. To obtain more information about the `comsol mpd` commands, add the `-h` option to the commands, for example, `comsol mpd boot -h`.

Start MPD on a single computer with the command

```
comsol mpd mpd &
```

This is useful when running all computational nodes on a single multiprocessor computer or when you have difficulties attaching computational nodes because of firewalls. In the second case you can start an MPD on each node and attach them by specifying the main port and host. Use

```
comsol mpd trace -l
```

and

```
comsol mpd mpd --port <the port number reported> --host <the    hostname reported>
```

You can also start COMSOL with the `-clustersimple` option. This option automatically starts and terminates the MPD daemon. It uses the `mpd.hosts` file in your home directory to determine what computational nodes to use.

Starting Distributed COMSOL—Linux Examples

Make sure that COMSOL is able to start on all nodes where you intend to run COMSOL.

! Each node requires access to the license manager. If the node is unable to check out a license, it aborts the startup process.

A simplified version is used when the `-clustersimple` switch is set or the Hydra launcher is used. An example follows. Start four computational nodes on hosts listed in the file `hosts` using distributed COMSOL and simplified start:

```
comsol -nn 4 -clustersimple -f hosts alternatively
comsol -nn 4 -clustersimple -f hosts server alternatively
comsol -nn 4 -clustersimple -f hosts batch -inputfile in.mph -outputfile out.mph
```

where `-clustersimple` is optional for Hydra.

If you use MPD, start an MPD on a single computer. Then start distributed COMSOL on two computational nodes (on the same host) each using three processors, and finally stop the MPD:

```
comsol mpd mpd &
comsol -nn 2 -np 3 -mpd alternatively
comsol -nn 2 -np 3 -mpd server alternatively
comsol -nn 2 -np 3 -mpd batch -inputfile in.mph -outputfile out.mph
comsol mpd allexit
```

The example above could be used if you have a very small model with a very large amount of parametric steps, where using `mpd` on a single computer might be beneficial.

Start three MPDs on the compute nodes with hostnames defined in the file `myhosts`. Each line in the file should specify the host address or IP-address of a node. Make sure the MPDs were correctly booted. Then start a distributed COMSOL server on three computational nodes, and finally stop the MPDs. First make sure that you can connect to all the computers with `ssh` without having to use your password (see the man pages for `ssh`). Also make sure that all computers have access to the same COMSOL installation and that they are using the same Linux version. There are two options for starting a session. One more detailed and one shorthand version. An example of the detailed version:

```
comsol -nn 4 mpd boot -f myhosts
comsol mpd trace
comsol -nn 4 -mpd alternatively
comsol -nn 4 -mpd server alternatively
comsol -nn 4 -mpd batch -inputfile in.mph -outputfile out.mph
comsol mpd allexit
```

MPI Options

There are several implementations of MPI. COMSOL is shipped with the Intel MPI library but also supports most MPI implementations based on MPICH2. It is recommended that you use the default Intel MPI library. For running COMSOL on a computer that has MPICH2 installed, COMSOL also has a compatibility mode that you can activate by adding the option `-mpi mpich2`. When using this option both the variables `PATH` and `LD_LIBRARY_PATH` must include your MPI implementation. It is also possible to use other MPI libraries based on MPICH2 using the option `-mpipath` *`<path to shared library>`* and `-mpiroot` *`<path to root of mpi library installation>`*. Table 21-14 lists the MPI related options, `-mpi`, `-mpipath`, `-scalapack`, and `-scalapackpath`. Additionally the comsol MPI arguments are configurable inside the COMSOL start script. To configure COMSOL to work with a job scheduler through the Cluster Computing study you can set the options

```
-Dcs.precmd=<Command line>
-Dcs.postcmd=<Command line>
```

in the `comsol.ini` file. This adds commands prior to the `comsol` command and after the `comsol` command. You can add `{nn}` or `{perhost}` to any of these pre- or postcommands. This configures the Cluster Computing study to use the number of nodes and number of nodes on each host from the corresponding settings for the Cluster Computing study. For more information, see Cluster Computing.

Troubleshooting Distributed COMSOL and MPI

The Hydra launcher is the main MPI environment. The syntax for Hydra commands is

```
comsol [<options>] hydra [<Hydra command>] [<target arguments>]
```

TABLE 21-23: COMSOL HYDRA COMMANDS

COMSOL MPD COMMANDS	DESCRIPTION
`cleanup`	Run mpicleanup command
`mpitest`	Run a distributed test program
`tune`	Run mpitune command

Use the `-h` switch for more information about each command.

COMSOL ships with the Intel MPI library but should be compatible with most MPICH2 compatible MPI libraries. To download the latest version of Intel MPI library runtime visit http://software.intel.com/en-us/intel-mpi-library. To run COMSOL with another version of Intel MPI or other MPI library set `-mpiroot` to the root path of the MPI library and. In case the downloaded library is not compatible with the version COMSOL uses (this should usually not be the case), also set `-mpipath` to the dynamically loaded library that should be used. The default of the Intel MPI library is to use `ssh` as communication protocol. If you require another communication protocol use the option `-mpibootstrap` *`<protocol>`*. If you are using a scheduler, the Intel MPI library is often able to detect the environments it is running from using the `-clustersimple` switch and you do not need to set up a hosts file.

- If you are using a PBS or Torque scheduler add `-mpiarg -rmk -mpiarg pbs` to the command line in order for Intel MPI to interpret the environment correctly. The Intel MPI library automatically tries to detect the best option for communication and uses InfiniBand if it detects it. To verify that COMSOL is using InfiniBand, check the output from the startup of COMSOL:, it should not mention TCP transfer mode.
- If you have problems running on a Myrinet network add the options `-mpiarg -mx` to the command line.
- If you have problems running on a Qlogic network, add the options `-mpiarg -psm` to the command line.

In some cases it helps if you combine the option with the environment variable `PSM_SHAREDCONTEXTS_MAX` set to `1`. You can control the fabrics used for communication with the option `-mpifabrics fabric1:fabric2` where `fabric1` is equal to `fabric2` or `fabric1` is `shm`.

If COMSOL aborts during start make sure that all nodes are able to access the license manager and that COMSOL can be started on each node when not running distributed. Sometimes there is additional information in the log files located in `$HOME/.comsol/v44/configuration/comsol/*.log`. If this does not help, start the MPI test program to make sure that the MPI library is working as it should using the following command:

```
comsol -nn <number of nodes> -f <host file> hydra mpitest
```

For more verbose information about the startup process when using Hydra, use `-mpiarg -verbose`, or set `-mpidebug` to a value greater than the default 4.

The MPD daemon can be used in several ways to troubleshoot problems with the comsol MPI environment. The syntax for MPD commands is

```
comsol [<options>] mpd [<MPD command>] [<target arguments>]
```

TABLE 21-24: COMSOL MPD COMMANDS

COMSOL MPD COMMANDS	DESCRIPTION
`boot`	Run mpdboot command
`mpd`	Run mpd command
`exit`	Run mpdexit command
`allexit`	Run mpdallexit command
`cleanup`	Run mpdcleanup command
`trace`	Run mpdtrace command
`check`	Run mpdcheck command
`ringtest`	Run mpdringtest command
`listjobs`	Run mpdlistjobs command
`sigjob`	Run mpdsigjob command
`killjobs`	Run mpdkilljobs command
`mpitest`	Run a distributed test program
`tune`	Run mpdtune command
`help`	Run mpdhelp command

Use `-h` switch for more information about each command.

When using MPD, use the `comsol mpd check` command to display important information. For more verbose information about the startup process from the MPD daemon, use the `-v` and `-d` switches, set the environment variable, or set `-mpidebug` to a value greater than the default 4. If the MPD is booted and COMSOL is not starting, make sure that the MPI environment is working by running the `comsol mpd mpitest` command, for instance.

COMSOL MATLAB COMMAND

Use the COMSOL `matlab` command to access the COMSOL API through MATLAB®. Enter:

```
comsol server matlab [<options>]
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsol server matlab` command:

TABLE 21-25: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
`-mlroot <path>`	MATLAB installation directory
`-host <hostname>`	Connect to host
`-port <hostname>`	Connect to port
`-desktop`	Start with Desktop
`-nodesktop`	Start without Desktop

TABLE 21-25: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
`-mlnosplash`	Start without MATLAB splash screen
`-graphics`	Start the server with graphics libraries. This enables plotting on the server. Available only when running `comsol server matlab [<options>]`.

COMSOL Commands on Macintosh

Use the COMSOL command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL command is

```
comsol [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. The `comsol` command can be combined with optional targets to achieve various results. The following table lists the command and targets:

TABLE 21-26: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
`comsol`	Run standalone COMSOL Multiphysics	
`comsol server`	Start COMSOL Multiphysics server	
`comsol client`	Run COMSOL Multiphysics client	
`comsol batch`	Run a COMSOL MPH file or class file	
`comsol compile`	Compile a Model Java file	
`comsol server matlab`	Start MATLAB and connect to a COMSOL server	Requires LiveLink™ *for* MATLAB® license
`comsol convertpre35a`	Convert 3.0–3.5 models	

The `comsol` command is located in the `bin` folder in the COMSOL installation directory.

INI FILES

There is a number of `.ini` files in the subdirectories `maci32` and `maci64` in the `bin` directory. It is sometimes recommended that you edit these files. For example, you can add options to any of the above commands by modifying the corresponding INI file. To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsol batch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the COMSOL command and target. Table 21-14 lists the options (see `[<options>]` in the command syntax) available for all comsol commands. Always issue these options between the command and the target (if any).

TABLE 21-27: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-h`	Print general help.	
`<target> -h`	Print target-specific help.	

TABLE 21-27: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
`-3drend ogl` | `sw`	3D renderer: OpenGL or software rendering.	
`-docroot` *`<path>`*	Specify custom path to the COMSOL documentation root directory.	See Documentation and Model Libraries Root Directories
`-modelsroot` *`<path>`*	Specify custom path to the COMSOL Model Libraries root directory.	See Documentation and Model Libraries Root Directories
`-np` *`<no. of processors>`*	Number of processors.	See Shared-Memory Options
`-mpmode throughput` | `turnaround` | `owner`	Multiprocessor mode.	See Shared-Memory Options
`-blas {auto}` | `mkl` | `acml` | `path`	BLAS library to use.	See BLAS Options
`-blaspath` *`<path>`*	BLAS library path.	See BLAS Options
`-ipv6`	Activate IPv6 support.	
`-c` *`<path>`*	License file path.	
`-prefsdir` *`<path>`*	Preference directory.	
`-tmpdir` *`<path>`*	Temporary file directory.	
`-version`	Print COMSOL version.	
`-version` *`<target>`*	Print target version.	
`-ckl`	Use classkit license.	
`-autosave <{on}`|`off>`	Control saving of recovery files.	
`-recoverydir` *`<path>`*	Path to recovery directories.	

For the `-tmpdir` option, COMSOL uses the specified directory to store temporary files. Use the `-prefsdir` option to specify the directory where COMSOL stores the preference file.

Documentation and Model Libraries Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Libraries root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Libraries root directories can be useful for administering Model Library Update; see The Model Libraries Window.

Setting the paths to the documentation and Model Libraries root directories using these options does not in itself move the directories and their contents.

Shared-Memory Options

Use the option `-np` to control the number of core and processors used. The default is to use all available cores and processors.

Depending on how loaded your machine is, you can control how COMSOL uses the available processors. The following options are available:

TABLE 21-28: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
`throughput`	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
`turnaround`	Typically provides the best performance when no other processes than COMSOL are active.
`owner`	Provides the highest performance in most cases.

Sometimes you might want to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides the following BLAS-related options:

TABLE 21-29: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL.
mkl	Use the Intel MKL library.
path	Use a BLAS library specified using the option `-blaspath` or the environment variable `COMSOL_BLAS_PATH`.

MKL is distributed along with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in `/lib/ARCH`, where `ARCH` is the architecture (`maci32` or `maci64`), or somewhere in the standard search path. You must also provide the path to any sublibraries needed by the library. You can also set the search path to point to the directory where the library is installed. To do so, use the environment variable `DYLD_LIBRARY_PATH`. Your library must support both the standard FORTRAN LAPACK interface and the standard FORTRAN BLAS interface. If your LAPACK and BLAS interface consists of several libraries, use the path to the LAPACK library.

COMSOL COMMANDS

In additions to the options in Table 21-14, the standalone COMSOL command supports the following option:

TABLE 21-30: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
`-open <file>`	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ *for* MATLAB® also needs to connect to a COMSOL server.

The syntax for the COMSOL server command is

```
comsol [<options>] server [<target arguments>]
```

The following target arguments are available for a COMSOL server command:

TABLE 21-31: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTIONS	DESCRIPTION
`-user <user>`	Specify login name for a user.
`-port <port>`	Specify a TCP/IP port to listen for connect attempts.
`-passwd reset|nostore`	Specify that you want to provide a new password.To avoid storing the new password on file use <nostore>.
`-login {info}|force|never|auto`	Ask for login information. `info` means that only missing information is asked for. `force` resets the password. `never` requires that the login information is available. `auto` automatically creates a new username and password.
`-multi on|{off}`	Accept repeated client connections.
`-silent`	Do not listen to standard input.

Accessing the COMSOL Server Computer

To access the computer running the COMSOL server simply log in on the server computer by using `ssh` or a similar command, then enter the `comsol server` command.

Login Information

When you start a COMSOL server for the first time, you are asked for a username and password. Select a username and a password, which COMSOL then uses in communications between the COMSOL client and the server. You must also specify a matching username and password on the settings page in the **Model Navigator**, which opens when you start the COMSOL client. The software writes this login information in the file `login.properties`. The login information is located in `Library/Preferences/COMSOL/v44/login.properties` in your home directory.

Client/Server Security Issues

COMSOL can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.

> ! Always make sure that untrusted users cannot access the COMSOL login information. Protect the file `Library/Preferences/COMSOL/v44/login.properties`. This is important when using the COMSOL client/server configuration. Alternatively, start the COMSOL server with the `-passwd nostore` option, and clear **Remember Password** when connecting to the server. This ensures that your login information is not stored on file.

Once you start a COMSOL server, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the COMSOL server, both in your firewall and by changing the COMSOL server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>/bin/conf/server.xml` and find the lines:

```
<!-- To restrict access to the COMSOL server you can uncomment the block below.
```

and follow the instructions. The default port for the COMSOL server is 2036. You can change this by using the option `-port <port>` when launching COMSOL and COMSOL server.

Documentation Security Issues

To serve the COMSOL Desktop with documentation COMSOL opens a separate documentation server on the client computer when you open the documentation.

To enhance security, you can limit the address range that can access the server by limiting the address range that can access the documentation server, both in your firewall and by changing the documentation server configuration. To limit the allowed address range in the server, edit the file `<COMSOL Installation Directory>/doc/help/conf/server.xml` and find the lines:

```
<!-- To restrict access to the documentation server you can uncomment the block below.
```

and follow the instructions. The default port for the documentation server is 8090. You can change this by using the option `-docport <docport>` when launching COMSOL.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with a the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command.

TABLE 21-32: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
`-port <port>`	Specify a TCP/IP port to connect to
`-server <server name>`	Specify server to connect to
`-open <file>`	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. You can run both Model MPH-files and model files for Java with the COMSOL batch command. Model files for Java need to be compiled before running.

The syntax for the COMSOL batch command is

```
comsol [<options>] batch [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-33: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
`-inputfile <file name>`	Run a Model MPH-file or class file.
`-outputfile <file name>`	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output.
`-job <job tag>`	The batch job to run.
`-study <study tag>`	The study to compute.
`-pname <parameter name>`	Comma-separated list of parameter names.

TABLE 21-33: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
`-plist <parameter value>`	Comma-separated list of parameter values.
`-batchlog <file name>`	File to store log in.
`-client`	Run as client.
`-host`	Connect to host.
`-port`	Connect to port.
`-checklicense <filename>`	Print license requirements for a Model MPH-file.
`-nosave`	Do not save the resulting model.

Example

To use COMSOL in batch mode to solve a model, run the following command:

```
comsol batch -inputfile in.mph -outputfile out.mph -study std1
```

This command starts COMSOL in batch mode, solves the model in the Model MPH-file with the given file name using the active solver settings in the model, and stores the solution in the `out.mph`.

The `-study` option directs COMSOL to run a certain study. The study is identified by its tag. In the COMSOL Desktop, select **Show Name and Tag** under **Model Builder Node Label** to see the tags of the jobs under **Study** within curly braces in the Model Builder. In the model object, determine the tags of the jobs by the command `model.study().tags()`. You can determine the name of each study by `model.study(<tag>).name()` using one of the job tags.

The `-job` option works similar to the `-study` option. It directs COMSOL to start a certain job. The job is identified by its tag. In the model object, determine the tags of the jobs by the command `model.batch().tags()`. You can determine the name of each job by `model.batch(<tag>).name()` using one of the job tags.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a model file for Java for use by the COMSOL batch command or for loading class files into the GUI. The syntax for the COMSOL compile command is

```
comsol [<options>] compile [<target arguments>] <file>.java
```

The Java file is mandatory. The following optional target arguments are available‘

TABLE 21-34: COMSOL COMPILE OPTIONS

COMSOL COMPILE OPTIONS	DESCRIPTION
`-jdkroot <path>`	Path to the JDK root
`-classpathadd <classpath>`	Additional classpath
`-verbose`	Verbose output

COMSOL MATLAB COMMAND

Use the COMSOL MATLAB command to access the COMSOL API through MATLAB. Type:

```
comsol server matlab [<options>]
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsol server matlab` command:

TABLE 21-35: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
`-mlroot <path>`	MATLAB installation directory
`-host <hostname>`	Connect to host
`-port <hostname>`	Connect to port
`-desktop`	Start with Desktop
`-nodesktop`	Start without Desktop
`-mlnosplash`	Start without MATLAB splash screen

The COMSOL Convertpre35a Command

Use the `comsol convertpre35a` command to convert a directory with models made in COMSOL 3.0–3.5 to COMSOL 3.5a. To use the command, enter

```
comsol [<options>] <input directory> <output directory> [<logfile>]
```

where *`<input directory>`* is the input directory, *`<output directory>`* is the output directory, and `[<logfile>]` is an optional log file. If you do not provide the third argument, the log is printed on standard output.

TABLE 21-36: COMSOL CONVERTPRE35A OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
`-c35aroot <path>`	Installation path of COMSOL 3.5a

22

Glossary

This Glossary of Terms contains terms related to finite element modeling, mathematics, geometry, and CAD as they relate to the COMSOL Multiphysics® software and documentation. For more application-specific terms, see the glossaries in the documentation for most of the add-on modules. For references to further information about a term, see the index.

Glossary of Terms

adaptive mesh refinement A method of improving solution accuracy by adapting the mesh to the problem's physical behavior.

affine transformations Geometric transformations that are combinations of linear transformations and translations.

algebraic multigrid (AMG) An *algebraic multigrid* solver or preconditioner performs one or more cycles of a multigrid method using a coarsening of the discretization based on the coefficient matrix. Compare to *geometric multigrid (GMG)*.

anisotropy Variation of material properties with direction.

application program interface (API) An *API* provides a set of documented functions and methods for interacting with a software product.

arbitrary Lagrangian-Eulerian formulation (ALE formulation) A formulation where an Eulerian equation is transformed into an equation written with respect to a mesh, which can be moving in relation to both the Eulerian frame and the Lagrangian frame. The COMSOL Multiphysics solvers have built-in support for the necessary transformation of derivatives.

arc A segment of the circumference of a circle or ellipse.

Argyris element A 2D, 6-node triangular finite element with a 5th-order *basis function* providing continuous derivatives between elements.

aspect ratio The ratio between the longest and shortest element or geometry dimension.

assemble Taking the local element stiffnesses, masses, loads, and constraints to form the *stiffness matrix*, *mass matrix*, load vector, constraint matrix, and constraint residual vector.

associative geometry An algorithm that maps data associated with a geometry to the new *geometric entities* when the geometry is modified.

backward differentiation formula (BDF) A multistep formula based on numerical differentiation for solutions to *ordinary differential equations*. A BDF method of order n computes the solution using an nth-grade polynomial in terms of backward differences.

basis function A function φ_i in the *finite element space* such that the ith degree of freedom is 1, while all other degrees of freedom are 0. For the Lagrange finite element space, φ_i is a linear or higher order polynomial on each mesh element with value 1 in node i and 0 in all other nodes.

Bernstein polynomial See *Bézier basis*.

Bézier basis A set of polynomial functions that occur in the definition of a *Bézier curve*. These polynomial functions are often called *Bernstein polynomials*.

Bézier curve A *rational Bézier curve* is a parameterized *curve* formed as the quotient of two polynomials expressed in the Bézier basis. It is a vector-valued function of one variable. The coefficients of a rational Bézier curve are geometrically interpreted as *control points* and *control weights*. A *nonrational Bézier curve* is a rational Bézier curve with all weights equal, thereby making the denominator polynomial equal to a constant. A nonrational Bézier curve is also called an *integer Bézier curve*.

Bézier patch, Bézier surface A *Bézier patch* or *Bézier surface* is a surface extension of a *Bézier curve*. A *Bézier patch* is a function of two variables with an array of control points.

bidirectional constraint A constraint enforced by reaction terms affecting both equations in a constraint of the type $u_1 = u_2$. *Symmetric constraints* are an important special case. See also *reaction terms* and *constraint*.

Boolean operations Boolean operations are used to construct a *geometry object* from other geometry objects. At least two primary geometry objects are required to create a resultant new geometry object. That new object depends on the type of Boolean operation:

- Union (add): the resultant geometry object occupies all the space of the initial geometry objects
- Difference (subtract): the resultant geometry object occupies all the space of the first geometry object except for the space inside the second geometry object.
- Intersection: the resultant geometry object occupies only the space common to the initial geometry objects

boundary A *geometric entity* with a dimension one less than the space dimension for the geometry (a *face* in a 3D geometry, an *edge* in a 2D geometry, and a *vertex* in a 1D geometry). In a mathematical context, the symbol $\partial\Omega$ represents the boundary of the domain Ω. Sometimes *boundary* is used in a narrower sense meaning an *exterior boundary*. See also *interior boundary*, *exterior boundary*.

boundary element method (BEM) A computational method that provides an alternative to the finite element method in case of a source free stationary PDE with constant coefficients.

boundary modeling A geometry modeling method to create a geometry by defining its boundaries. Compare to *solid modeling* and *surface modeling*.

brick element See *hexahedral element*.

chamfer A CAD operation that trims off a corner with a plane or straight line.

Cholesky factorization A memory-saving version of *LU factorization* where U is the transpose of L. It requires that the coefficient matrix A ($A = LU$) be a symmetric positive definite matrix. See also *LU factorization* and *positive definiteness*.

coefficient form PDE A PDE in the coefficient form is a PDE formulation suited for linear PDEs.

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu & \text{on } \partial\Omega \\ hu = r & \text{on } \partial\Omega \end{cases}$$

component coupling User-defined component couplings are used to couple data within a model component (geometry) or between different model components (geometries). See also *extrusion component coupling*, *projection component coupling*, and *integration component coupling*. Component couplings can be reused with different arguments (for example, for integrating different quantities over the same domain). Component coupling used to be called model coupling.

composite geometry object Geometric objects made up by combining *primitive geometry objects* and other composite objects. See also *constructive solid geometry*, *primitive geometry object*, and *Boolean operations.*

COMSOL Desktop The integrated simulation environment for the COMSOL products with a number of windows such as the Model Builder window, the Graphics window, and each model tree node's settings window.

COMSOL binary file A binary data file with the extension `.mphbin` that contains geometry objects or mesh objects.

COMSOL text file A text data file with the extension `.mphtxt` that contains geometry objects or mesh objects.

COMSOL server The COMSOL server is a single user server allowing multiple sessions of the same user, one session at a time.

condition number A measure of the possible error in a solution due to ill-conditioning of the equations. See also *ill-conditioning.*

constant A named model property that has a constant numeric value. The built-in constants in COMSOL include *mathematical and numerical constants* and *physical constants.*

constraint Restriction imposed upon the dependent variables on the form $R(u_1,u_2,...) = 0$. A *Dirichlet boundary condition* is a special case. *Neumann boundary conditions* are not regarded as constraints. When a constraint is added, the finite element algorithm adds corresponding *reaction terms* to the system of equations. These generalized *reaction forces* modify the *flux conditions* so that the resulting model becomes solvable.

constructive solid geometry (CSG) A solid-modeling method that combines simple solid shapes, or *primitives*, to build more complex models using Boolean operations. See also *solid modeling* and *primitive.*

contributing node A boundary condition or source is *contributing* when it adds to other boundary conditions or sources defined on the same geometric entity. Examples of contributing boundary conditions are loads in structural mechanics and heat flux components in heat transfer. See also *exclusive nodes.*

control point *Bézier* and *NURBS* curves and surfaces are defined by a set of points known as *control points.* The locations of these points control the curve's shape.

control weight Scalar values assigned to *control points* to further control the shape of a curve or surface.

contour plot A plot that shows the variation of a solution component or other quantity. Points with equal values of the plotted quantity are connected with contour lines.

convergence The tendency for a finite element solution to approach the exact solution within well-defined and specified tolerances, for example, by reducing the mesh element size or the time step.

curl element See *vector element.*

curve The path of a point moving through space. See also *Bézier curve*, *NURBS*, and *manifold.*

curve object A geometry object consisting of only *edges* and *vertices* (where no vertex is isolated), for example, a geometry object representing a *curve.*

curve segment An individual polynomial or rational polynomial curve. Compounded curves consist of several *curve segments.*

degree of freedom (DOF) One of the unknowns in a discretized finite element model. A degree of freedom is defined by a name and a *node point*. The degree of freedom names often coincide with the names of the dependent variables. The local degrees of freedom are all degrees of freedom whose node points are in one mesh element.

deformed geometry A geometry where the shape changes with a moving-mesh algorithm. It is also the name of a *physics interface* for modeling deforming geometries. This is similar to the Parameterized Geometry interface in earlier versions of COMSOL.

deformation gradient In solid mechanics, it contains the complete information about the local straining and rotation of the material. It is a nonsingular matrix with positive determinant, as long as material cannot be annihilated.

dependent variable A varying quantity whose changes are arbitrary but regarded as produced by changes in other variables that the varying quantity depends on. For example, temperature is a function of the space coordinates and time. In a narrower sense, the dependent variables, or *solution components*, are the unknowns in a mathematical PDE model. Compare to *independent variable*.

differential-algebraic equation (DAE) A set of equations that includes both differential and algebraic equations. A DAE is classified in terms of its *index*, a positive integer, which is related to the minimum number of differentiations needed to transform a DAE to an ODE form.

direct solver A solver for a system of linear equation that uses some variant of Gaussian elimination. Compare to *iterative solver*.

Dirichlet boundary condition A Dirichlet boundary condition specifies the value of the function (dependent variable) on a boundary. Dirichlet boundary conditions are sometimes called *essential boundary conditions* or *constraints*. See also *constraint*.

discretization The process of dividing a continuous system into a finite number of elements with finite size. The difference between the finite-element representation and the real system, the discretization error, drops as the size of the elements decrease. For a time-dependent analysis, a discretization of time into steps provides an idealized behavior of the variations in the solution during these steps.

divergence element A finite element with properties suitable for representing certain electromagnetic vector fields. The degrees of freedom on the boundary of a mesh element correspond to normal components of the field.

domain A topological part of the modeling space in a geometry model. The geometric representation of a domain is a line segment (interval) in 1D, an area in 2D, and a volume in 3D. In a mathematical context, the symbol Ω represents the domain where the equations are defined.

domain decomposition Domain decomposition is a solver method that divides the modeling domain into subdomains where the equations in the subdomains are easier to solve. The total solution is then obtained by iterating between the computed solutions for each subdomain using the currently known solutions from the other subdomains as boundary conditions.

drop tolerance A nonnegative scalar used in the incomplete LU preconditioner for the iterative solvers. See *incomplete LU factorization*.

dynamic model See *time-dependent model*.

edge, edge segment A *geometric entity* representing a bounded part of a *curve*. An *edge* or *edge segment* is a *boundary* in a 2D geometry. See also *domain*.

edge element See *vector element*.

eigenvalue PDE A PDE that describes an eigenvalue problem with unknown eigenmodes (eigenfunctions) u and eigenvalues λ. The *coefficient form* eigenvalue PDE is:

$$\lambda^2 e_a u - \lambda d_a u + \nabla \cdot (-c\nabla u - \alpha u) + \beta \cdot \nabla u + a u = 0$$

elliptic PDE A linear stationary 2nd-order elliptic PDE has the form

$$\nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

where c is positive or negative definite, for example, Poisson's equation.

embed To insert a 2D geometry into a 3D geometry model.

error Deviations from the correct solution, primarily due to: poor modeling; *discretization* (such as insufficiently fine mesh, poor elements, or insufficiently short time steps); and roundoff and truncation (depending on numerical representation, *ill-conditioning*, or the solution algorithms).

error estimate An estimation of the error in the numeric solution to a problem, either locally or globally, primarily for use by an adaptive mesh refinement. See also *adaptive mesh refinement*, *error*.

equivalent boundaries *Boundaries* that are rigid transformations of each other and have compatible meshes. See also *periodic boundary condition*.

essential boundary condition See *Dirichlet boundary condition*.

Eulerian formulation An Eulerian formulation means that the partial differential equations that describe some physics are formulated in a *spatial frame* (coordinate system), with coordinate axes fixed in space. An Eulerian formulation is common for fluid flow when the focus is on specific locations in space through which fluid flows. Compare to *Lagrangian formulation*.

exclusive node A boundary condition or material model in a domain is *exclusive* when there can only be one such node defined for a given geometric entity. Adding another exclusive boundary condition to the same boundary, for example, the last added boundary condition (last in the Model Builder tree) overrides any other similar boundary condition defined on the same boundary. Examples of exclusive boundary conditions are prescribed displacements in structural mechanics and specified temperature in heat transfer. See also *contributing node*.

extended mesh A data structure that includes the full finite element mesh. See also *mesh*, *node point*.

extended multiphysics A model that includes nonlocal couplings and dependencies between variables, where the value at a point is the result of a computation elsewhere in the domain or in another geometry defined in the same model. *Coupling operators* provide the ability to project or extrude values from one geometry or domain to another. Compare to *multiphysics*.

exterior boundary An *exterior boundary* for a dependent variable u is a *boundary* such that u is defined only on one of the adjacent domains, that is, a boundary to the computational domain. See also *boundary*.

extrude To create a 3D geometry object from a 2D geometry object in a *work plane* or a planar face in 3D by translating (extruding) it in the normal direction.

extrusion component coupling A coupling defined in the destination that takes values from the source by interpolation at points that depend on the position of the evaluation points in the destination.

face A *geometric entity* describing a bounded part of a *surface* in a 3D geometry. A *face* is a *boundary* in a 3D geometry. See also *domain*.

fallback feature Used with the pair node to enable pairs to have the option to add additional subnodes with conditions for nonoverlapping parts of the pair.

FEM See *finite element method*.

Fick's law The first law relates the concentration gradients to the diffusive flux of a solute infinitely diluted in a solvent. The second law introduces the first law into a differential material balance for the solute.

field variables Dependent variables and variables derived from them. Compare to *expression variables.*

fillet A curved transition from one boundary to another, creating a rounded corner.

finalized geometry The resulting geometry used for assigning materials and physics. COMSOL creates the finalized geometry by forming a union of the entire *geometry sequence* or by forming an assembly where the geometry objects in the geometry sequence are treated as individual parts. The finalized geometry consists of *geometric entities.*

finite element In the mathematical sense, a *mesh element* together with a set of *shape functions* and corresponding *degrees of freedom*. The linear combinations of the shape functions form a space of functions called the *finite element space*. In the traditional FEA sense, the concept of a finite element also includes the discretized form of the PDEs that govern the physics. COMSOL generally uses *finite element* in the mathematical sense.

finite element analysis (FEA) A computer-based analysis method for field problems using the *finite element method*.

finite element method (FEM) A computational method that subdivides an object into very small but finite-size elements. The physics of one element is approximately described by a finite number of *degrees of freedom (DOFs)*. Each element is assigned a set of characteristic equations (describing physical properties, boundary conditions, and imposed forces), which are then solved as a set of simultaneous equations to predict the object's behavior.

finite element space The linear space of functions where the finite element approximation to the solution of a PDE problem is sought. The functions in the finite element space are linear combinations of *basis functions* (*shape functions*).

finite volume method (FVM) A computation method that, in way similar to the *finite element method*, computes values at discrete places on a meshed geometry. Finite volume refers to the small volume surrounding each node point in a mesh.

flux condition A boundary condition which specifies the value of the *normal flux* across a boundary, also known as a *natural boundary condition*. A (generalized) *Neumann boundary condition* is a special case.

flux vector The general flux vector is as below, with three terms: the first term describes diffusion, the second term describes convection with a velocity $-\alpha$, and the third term γ is a source term. See also *generalized Neumann boundary condition* and *normal flux.*

$$\Gamma = -c\nabla u - \alpha u + \gamma$$

frame A *frame* is a coordinate system that is fixed in space, to a material, to the geometry, or to a mesh. The frames make it possible to use an *Eulerian formulation* or a *Lagrangian formulation* for various physics in a model or using the arbitrary Lagrangian-Eulerian (ALE) method. The following frame types are available: *material frame* (*reference frame*), *geometry frame*, *mesh frame*, and *spatial frame*.

free mesh An *unstructured mesh* that can represent any geometry. Compare to *mapped mesh*.

free mesher The mesh generator creating *free meshes*. The mesh generator creating *triangular elements* is also referred to as the *free triangle mesher*, and the mesh generator creating *quadrilateral elements* is also referred to as the *free quad mesher*.

free quad mesher The mesh generator creating unstructured quadrilateral meshes.

free tet mesher The mesh generator creating unstructured tetrahedral meshes.

free triangle mesher The mesh generator creating unstructured triangular meshes.

function COMSOL supports *user-defined functions*, which can be analytic, piecewise, and interpolation functions as well as special types of common functions that implement, for example, steps, ramps, and other wave forms. There are also common built-in *mathematical functions* such as trigonometric functions, logarithms, and special functions.

Gauss point Sometimes improperly used as a synonym for *integration point*. See also *integration point*.

general form PDE A PDE in the general form is a PDE formulation suited for nonlinear PDEs

$$\begin{cases} e_a \dfrac{\partial^2 u}{\partial t^2} + d_a \dfrac{\partial u}{\partial t} + \nabla \cdot \Gamma = F & \text{in } \Omega \\ -\mathbf{n} \cdot \Gamma = G + \left(\dfrac{\partial R}{\partial u}\right)^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

generalized Neumann boundary condition A generalized Neumann boundary condition (also called a *mixed boundary condition* or a *Robin boundary condition*) specifies the value of a linear combination of the *normal flux* and the *dependent variables* on a boundary. For a coefficient form PDE, the generalized Neumann boundary condition is

$$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g - h^T \mu$$

The generalized Neumann condition is often called just *Neumann condition* in the documentation.

generalized reaction force see *reaction term*.

geometric entities The basic parts that constitute the finalized geometry: In 3D they are divided in the following four types or *geometric entity levels*: *domains*, *boundaries* (*faces*), *edges*, and *points* (*vertices*). In 2D, there are no *faces*, and the *edges* are the *boundaries*. In 1D there are only *domains* and *points*, which are also the *boundaries*.

geometric multigrid (GMG) A *geometric multigrid* solver or preconditioner performs one or more cycles of a multigrid method, using a coarsening of the discretization based on a coarsening of the mesh or a reduction in the order of the shape functions. Compare to *algebraic multigrid (AMG)*.

geometry frame In the geometry frame (coordinate system) the domain is fixed and identical to the original geometry. No physics is formulated directly in the geometry frame—only the *material frame* and *spatial frame* have physical significance. The geometry frame is used only as a reference for the Deformed Geometry interface and for postprocessing. When there is no Deformed Geometry interface present, the geometry frame is identical to the material frame.

geometric entity level The *geometry entity levels* are the *vertex*, *edge*, *face*, and *domain* levels. An entity of dimension one less than the space dimension is referred to as a *boundary*. See also *geometric entities*.

geometry model A collection of *geometric entities* that form a complete geometric description of the model.

geometry object An object generated by a geometry feature. See also *point object*, *curve object*, *surface object*, *primitive geometry object*, *solid object*, and *mixed object*.

geometry sequence The sequence of geometry features that define a geometry (of a model component) plus other settings that define the geometry. In the Model Builder, this is represented by the Geometry node and its child nodes.

grid A *grid* usually refers to sets of evenly-spaced parallel lines at particular angles to each other in a plane, or the intersections of such lines. Compare to *mesh*.

Hermite element A finite element similar to the *Lagrange element*. The difference is that there are degrees of freedom for the (1st-order) space derivatives at the mesh vertices. See also *Lagrange element*.

hexahedral element A 3D mesh element with eight corners and six faces, also referred to as *brick element*; sometimes also called *hex element* as a short form.

higher-order element A finite element with *basis functions* that consists of polynomials of degree 2 or higher.

hybrid geometry modeling Creating a geometry model using a combination of *boundary modeling/surface modeling* and *solid modeling*.

hyperbolic PDE A typical example of a linear 2nd-order hyperbolic PDEs is the *wave equation*

$$e_a\frac{\partial^2 u}{\partial t^2} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f$$

where e_a and c are positive.

IGES file An *IGES file* contains 3D CAD data, including the 3D geometry, in an open format according to the *Initial Graphics Exchange Specification*. IGES files can be imported into COMSOL using the CAD Import Module.

ill-conditioning An ill-conditioned system is sensitive to small changes in the inputs and is susceptible to roundoff errors. See also *condition number*.

imprint An imprint of the usually smaller boundary on the larger boundary that makes the parts in a *pair* match. An imprint inserts points on the boundary in 2D and creates edges on the boundary in 3D.

incomplete LU factorization An approximate *LU factorization* where small matrix elements are discarded to save memory and computation time. The *drop tolerance* is a relative measure of the smallness of the elements that should be discarded. See also *LU factorization*.

independent variable A variable that can cause variation in a second, *dependent variable*. The independent variables are most often space coordinates and time. Compare to *dependent variable*.

index, for DAE See *differential-algebraic equation.*

initial condition The starting values for the dependent variables in a time-dependent analysis and for *nonlinear iterations* or other iterative solvers.

integration component coupling A coupling that evaluates integrals of expressions over the source and returns a single scalar value when used in the destination, which for this type of component coupling is the entire model. Similar functionality is available to evaluate the average, minimum, and maximum values.

integration point See *numerical integration formula.*

interactive meshing Building a mesh in an incremental fashion where each meshing operation acts on a set of geometry domains.

interior boundary An *interior boundary* for a dependent variable *u* is a *boundary* such that *u* is defined on both adjacent domains or in no adjacent domain. See also *boundary.*

interval The domain between two vertices (points) in a 1D geometry. Also called a *domain.*

isoparametric element A finite element that uses the same *shape function* for the element shape coordinates as for the *dependent variables.*

isosceles triangle A triangle with at least two equal sides (and two equal angles).

iteration See *iterative solver.*

iterative solver A solver for a system of linear equations that uses an iterative method, calculating a sequence of more and more accurate approximations to the solution. Each step in this sequence is one *linear iteration*. This should not be confused with the Newtons iterations (*nonlinear iterations*) that occur in the solution of a nonlinear system of equations. Compare to *direct solver* and *nonlinear iteration*.

Jacobian matrix A matrix containing the first derivative of a vector-valued function of a vector variable. In particular, it is the derivative of the *residual vector* with respect to the *solution vector.* When used in this narrower sense, the term *stiffness matrix* is sometimes used.

Lagrange element A *finite element* with polynomial shape functions of a certain *order* (degree). The value of the function is used as the *degree of freedom*, and the node points are the *Lagrange points.*

Lagrange multiplier An extra dependent variable introduced in the *flux conditions* when a constraint is added. The Lagrange multiplier often has a physical meaning and an interpretation as a (generalized) *reaction force*. See also *constraint.*

Lagrange point In a mesh element, the *Lagrange points* of order k are the points whose local (element) coordinates are integer multiples of $1/k$. These points are used as node points for the *Lagrange element.* For example, the Lagrange points of order 1 are the corners of the mesh element.

Lagrangian formulation A Lagrangian formulation means that the partial differential equations that describe some physics are formulated in a *material frame* (coordinate system) with coordinate axes fixed to the material in its reference configuration and following the material as it deforms. The Lagrangian formulation is common for solid mechanics because it makes anisotropic material properties independent of the current spatial orientation of the material. Compare to *Eulerian formulation.*

linear iteration A step in a linear iterative solver. See *iterative solver.* Compare to *nonlinear iteration.*

linear PDE An equation where both sides are sums of a known function, the unknown functions, and their partial derivatives, multiplied by known coefficients that only depend on the *independent variables*. Other PDEs are called *nonlinear*.

LU factorization For a linear system of equations, a version of Gaussian elimination that produces a factorization $A = LU$ of the coefficient matrix, where L and U are the lower and upper triangular matrices, respectively. This makes it easy to quickly solve a number of systems with the same coefficient matrix. See also *direct solver*.

mapped mesh A *structured mesh* with *quadrilateral elements* generated by mapping using transfinite interpolation.

mapped mesher The mesh generator creating *mapped meshes*.

mass matrix The matrix E that multiplies the second time derivative of the *solution vector* in the linearized discretized form of a PDE problem. If there are no second time derivatives (that is, if $E = 0$), then the term mass matrix is often used for the matrix D that multiplies the first derivative of the solution vector (the D matrix is otherwise called the *damping matrix*).

material frame The material frame defines a coordinate system that is fixed to the material in its reference configuration and follows the material as it deforms. The material frame is used in connection with a *Lagrangian formulation*. This frame is also referred to as a *reference frame*.

mathematical and numerical constants Built-in common mathematical constants such as π and i and numerical constants such as the machine precision or machine epsilon.

mesh A subdivision of the entities of a geometric model into, for example, triangles (2D) or tetrahedrons (3D). These are examples of *mesh elements*. See also *grid*, *structured mesh*, and *unstructured mesh*.

mesh element The individual elements in the mesh that together form a partitioning of the geometry, for example, *triangular elements* and *tetrahedral elements*. See also *finite element*.

mesh frame In the mesh frame (coordinate system) the domain is fixed until an automatic or manual remeshing operation is performed, as well as between remeshing events. When remeshing is not used, the mesh frame is identical to the *geometry frame*.

mesh vertex An endpoint or corner of a mesh element. See also *node point* and *vertex*.

method of lines A method for solving a time-dependent PDE through a space discretization, resulting in a set of ODEs.

mixed boundary condition See *generalized Neumann boundary condition*.

mixed object A nonempty *geometry objec*t that is not a *solid object*, *surface object*, *curve object*, or *point object*. For example, the union of a solid object and a curve object is a mixed object.

mode reduction A model-reduction technique for reducing systems with many degrees of freedom, such as large finite element models, to a form with fewer degrees of freedom for dynamic system simulations and analysis. See also *state-space model*.

model coupling See *component coupling*.

model input *Model inputs* are fields such as temperature and velocities that act as inputs for materials and model equations. The model inputs can be fields computed by other physics interfaces or user-defined values.

model file for Java A file that contains Java® commands calling on the COMSOL *API*. Use a text editor to extend and modify the model file. Compiling and running a model file for Java creates the COMSOL model.

model file for MATLAB A text file containing commands that create a COMSOL model. A model file for MATLAB is a text file (M-file) that is similar to a *model file for Java* and that can be modified and used with MATLAB. If you have a MATLAB license and a license for LiveLink™ *for* MATLAB®, the COMSOL Desktop can load a model file for MATLAB. Compare with *Model MPH-file*.

Model MPH-file A binary data file with the extension `.mph` that contains a COMSOL model. Often also just called model file.

model object An object (data structure) that contains all data for a model. This is the fundamental data structure in a COMSOL model.

Model Wizard Part of the COMSOL Desktop that is used to start building a model. It contains the Select Space Dimension, Select Physics, and Select Study Type pages.

MRI data *Magnet resonance imaging (MRI) data* is an image data format, primarily for medical use. MRI produces high-quality images of the inside of the human body. 3D MRI data is usually represented as a sequence of 2D images.

multigrid A solver or preconditioner for a linear system of equations that computes a sequence of increasingly accurate approximations of the solution by using a hierarchy of coarsened versions of the linear system (having fewer degrees of freedom). See also *algebraic multigrid*, *geometric multigrid*.

multiphysics Multiphysics models include more than one equation and variable from different types of physics. These variables can be defined in different domains. The equations can be coupled together through equation coefficients that depend on variables from other equations. Compare to *extended multiphysics*.

natural boundary condition See *Neumann boundary condition*.

Neumann boundary condition A Neumann boundary condition specifies the value of the *normal flux* across a boundary. Neumann boundary conditions are sometimes called *natural boundary conditions*. Compare to *generalized Neumann conditions*.

Newton's method An iterative solver method, also called the *Newton-Raphson method*, for solving nonlinear equations. See also *nonlinear iterations*.

Newton-Raphson method See *Newton's method*.

node point Any point in the mesh element where the degrees of freedom are defined. The node points often include the mesh vertices and possibly interior or midpoint locations. See also *degree of freedom* (DOF) and *mesh vertex*.

nonlinear iteration A Newton step in the solution of a nonlinear PDE problem. Each nonlinear iteration involves the solution of a linear system of equations. Compare to *linear iteration*.

nonlinear PDE See *linear PDE*.

norm A scalar measure of the magnitude of a vector or a matrix. Several types of norms are used to measure the accuracy of numerical solutions.

numerical integration formula A numerical integration method that approximates an integral by taking the weighted sum of the integrand evaluated at a finite number of points, the *integration points* (sometimes improperly called *Gauss points*). Also called *quadrature formula*.

normal flux The normal component of the *flux vector* at a boundary.

NURBS The *nonuniform rational B-spline (NURBS)* is a curve and surface representation scheme. A NURBS representation can be divided into a number of *rational Bézier curves* or surfaces.

operator, operator function A user-defined *operator function*, or just *operator*, is similar to a *function* but behaves differently. For example, COMSOL includes differentiation operators that take expressions as input arguments to define a derivative of an expression with respect to a variable. There are also built-in arithmetic, relational, and logical operators.

order of a finite element The degree of the polynomials that define the *shape functions* (*basis functions*).

ordinary differential equation (ODE) An equation involving functions and their derivatives. The derivatives are with respect to one independent variable only. Compare to *partial differential equation (PDE)*.

parabolic PDE A typical example of a linear 2nd-order parabolic PDE is the *heat equation*

$$d_a\frac{\partial u}{\partial t} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f$$

where d_a and c are positive.

parameter A constant that can take on different values for each model in a parametric analysis. See also *constant*.

partial differential equation (PDE) An equation involving functions and their partial derivatives; that is, an equation that includes derivatives with respect to more than one independent variable. Compare to *ordinary differential equation (ODE)*.

periodic boundary condition A boundary condition where the values of the solution appear in a periodic pattern, typically so that the value of the solution on one boundary is equal to the value on another boundary. See also *equivalent boundaries*.

phasor A complex number or a vector of complex numbers representing a sinusoidally varying current or voltage.

physical quantity A quantity (quantifiable property) that can be used in the mathematical equations of science and technology.

physics interfaces Sets of physics nodes for different types of physics in the COMSOL Desktop environment. The physics interfaces (sometimes referred to as the *physics*) contain predefined equations and boundary conditions and a set of nodes for setting up models for that type of physics.

pivot Usually a value on the main diagonal of the *stiffness matrix*. *Pivoting* is the interchanging of rows and columns in order to place a particularly large element in the diagonal position. The value of the diagonal element when it is used to eliminate values below it is called the *pivot value*.

point A location in space. Often used in a narrower sense with the same meaning as *vertex*.

point object A geometry object with only *vertices*.

positive definiteness A symmetric matrix is *positive definite* when all its eigenvalues are positive.

preconditioner The convergence rate of iterative methods depends on the spectral properties of the coefficient matrix. A *preconditioner* is a matrix that transforms the linear system into one that has the same solution but that has more favorable spectral properties. See also *algebraic multigrid*, *geometric multigrid*, *incomplete LU factorization*, *iterative solver*, and *SSOR*.

primitive, primitive geometry object A geometry object with a basic shape such as a cube or a sphere. Add primitives to a model, using arbitrary sizes and positions, and combine them to form complex shapes. See also *constructive solid geometry*, *composite geometry object*, and *Boolean operations*.

prism element A 3D mesh element with six corners and five faces, also referred to as *wedge element*.

projection component coupling A coupling that takes values from the source by evaluating line integrals over lines whose positions are dependent on the position of the evaluation points in the destination.

quadrature formula See *numerical integration formula*.

quadrilateral element A 2D mesh element with four corners and four edges; sometimes also called *quad element* as a short form.

rational Bézier curve See *Bézier curve*.

reaction force see *reaction term*.

reaction term Terms that are automatically added to the system of equations in order to enforce a *constraint*. Reaction terms from boundary constraints appear as a *flux condition* and share the same physical meaning. Using an analogy from structural mechanics, reaction terms are sometimes referred to as (generalized) reaction forces.

reference frame See *material frame*.

residual vector The vector L in the discretized form of a PDE problem. In the absence of *constraints*, the discrete form of a stationary equation is $0 = L(U)$ where U is the *solution vector*.

revolve To create a 3D geometry object from a planar face by rotating it about an axis.

Robin boundary condition See *generalized Neumann boundary condition*.

shape function A *basis function* described in local element coordinates. See also *basis function*.

shift A value σ around which an eigensolver searches for eigenvalues.

simplex element *Triangle element* in 2D and *tetrahedral element* in 3D.

solid See *solid object*.

solid modeling A 3D geometry modeling method that describes both the boundary and interior of the geometry using solid objects. See also *constructive solid geometry (CSG)* and *solid object*.

solid object A geometry object whose *vertices*, *edges*, and *faces* all have an adjacent *domain*.

solution component See *dependent variable*.

solution matrix A matrix that contains a sequence of solutions as columns. A steady-state problem results in a *solution vector*, but eigenvalue problems, time-dependent problems, and parametric analyses produce a *solution matrix.*

solution vector A vector with components that contain all the *degrees of freedom* (values of the *dependent variables*) as its components. See also *solution matrix.*

solver sequence A sequence of named solver settings and commands that can be replayed by a single solver call.

sparse matrix Matrix for which the number of zero elements is large enough to justify special data types and algorithms that avoid operations on zero elements.

spatial frame The spatial frame defines a coordinate system with coordinate axes fixed in space. The spatial frame (also called the Eulerian frame) is used in connection with an *Eulerian formulation*.

split To divide a geometry object into its minimal parts.

stability A solver for a time-dependent model is *unconditionally stable* if the initial conditions are not amplified artificially and the roundoff errors do not grow, regardless of the size of the time step. A solver is *conditionally stable* if there is a maximum value of the time step above which the numerical solution is unstable.

state-space model A linear time-invariant representation of a dynamic system as a set of 1st-order *ODEs* of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{y} &= Cx + Du \end{aligned}$$

where x is the state vector, u is the input, and y is the output. A, B, C, and D are the constant dynamics, input, output, and direct transmission matrices, respectively.

static model See *stationary model.*

stationary model A model where the dependent variables do not change over time. It typically represents a steady-state solution. Also called *static model* or *steady model.*

steady model See *stationary model.*

stiffness matrix See *Jacobian matrix.*

streakline The locus of particles that have earlier passed through a prescribed point in space. See also *streamline*.

streamline A curve that is tangent to the vector field everywhere (in particular a velocity field) at a given instant of time. Sometimes called a *flow line* or *flux line*. See also *streakline*.

streamline-diffusion stabilization A numerical technique for stabilization of the numeric solution to a PDE by artificially adding diffusion in the direction of the *streamlines.*

strong form A partial differential equation in the *strong form* is the standard formulation as an equality of functions. The strong form is divided into the *coefficient form* and the *general form*. Compare to *coefficient form*, *general form*, and *weak form*.

structured mesh A mesh for which all elements and nodes have the same topology. Compare to *unstructured mesh.*

surface A smooth mathematical function from 2D to 3D space.

surface normal A vector perpendicular to the surface.

surface modeling A 3D geometry modeling method to describe a geometry by defining its bounding surfaces. Compare *boundary modeling* and *solid modeling*.

surface object A geometry object without domains, isolated edges, or isolated vertices. Typically a trimmed surface is represented as a surface object.

swept mesh A 3D mesh generated by sweeping a face mesh along a domain.

symmetric matrix A matrix that equals its own transpose.

symmetric successive overrelaxation (SSOR) A *symmetric successive overrelaxation (SSOR)* preconditioner uses classic SSOR iterations.

symmetry The invariance of an object attribute or of the object itself under a transformation such as inversion, rotation, or reflection. A *symmetry* allows for a reduction of the model geometry so that appropriate boundary conditions account for the redundant portions of the geometry. Axial symmetry is a common type of *symmetry.*

symmetric constraint A *constraint* that is enforced by *reaction terms* chosen so as to preserve the symmetry of symmetric unconstrained systems. This choice of reaction terms is unique and leads to a *bidirectional constraint* that modifies the equations corresponding to all dependent variables appearing in the constrained expression.

symmetry boundaries See *equivalent boundaries.*

test function See *weak form.*

tetrahedral element A 3D mesh element with four corners, six edges, and four triangular faces.

time-dependent model See *transient model.*

transient model A model where at least one of the dependent variables changes over time, for example, the heat equation or the wave equation. Also called *dynamic model*, *time-dependent model*, or *unsteady model.*

triangular element A 2D mesh element with three corners and three edges.

trimmed surface If the parameter space of a surface is divided into “valid” and “invalid” regions, the image of the valid regions is called the *trimmed surface*. This corresponds to the part of the surface limited by a closed loop of edges lying on the surface.

unidirectional constraint A constraint enforced by *reaction terms* that only affect one of the dependent variables in a constraint of type $u_1 = u_2$. The other dependent variables are treated as independent with respect to the unidirectional constraint. Compare to *symmetric constraint*. See also *constraint.*

unstructured mesh A mesh without a specific pattern where the elements can have different shapes and the nodes can have different connectivities. Compare to *structured mesh.*

unsteady model See *time-dependent model.*

user-defined variable A user-defined variable can be defined on a global level or on any geometric entity in terms of *dependent variables*, *independent variables*, *parameters*, *constants*, and other *variables.*

vector element A finite element often used for electromagnetic vector fields. Each mesh element has degrees of freedom corresponding only to tangential components of the field. Also called *curl element*, *Nédélec's edge element*, or just *edge element*.

vertex A point in a geometry model, often an endpoint of an edge or an intersection of *geometric entities* of a higher degree such as *edges* or *faces*. A vertex is referred to as a *point* for the specification of point sources and other PDE modeling. See also *domain*.

weak constraint A reformulation of a *constraint* as a *weak form* equation. When using a weak constraint, the corresponding *Lagrange multiplier* becomes a *solution component* (*dependent variable*).

weak form A partial differential equation in the *weak form* is a more general formulation than the strong form. It is produced by multiplying the *strong form* PDE with an arbitrary function called the *test function* and integrating over the computational domain. Physics interfaces in COMSOL are implemented using a weak form. Compare to *strong form*.

wedge element See *prism element*.

well-posed A well-posed mathematical problem has a unique solution and depends continuously on its input, such as initial conditions, source terms, and boundary conditions.

work plane An embedded 2D work space that can be positioned relative to the coordinate planes or an already existing geometry. Using *work planes* makes it possible to define a geometry in terms of previously created geometry objects such as *points*, *edges*, and *faces*. From a work plane with a 2D geometry, 3D geometry objects can be created using *extrude* or *revolve* operations.

Index