COMSOL Multiphysics®

Reference Guide

VERSION 4.2a

Benelux

COMSOL BV
Röntgenlaan 37
2719 DX Zoetermeer
The Netherlands
☎ +31 (0) 79 363 4230
🖷 +31 (0) 79 361 4212
✉ info@comsol.nl
💻 www.comsol.nl

Denmark

COMSOL A/S
Diplomvej 381
2800 Kgs. Lyngby
☎ +45 88 70 82 00
🖷 +45 88 70 82 88
✉ info@comsol.dk
💻 www.comsol.dk

Finland

COMSOL OY
Arabiankatu 12
FIN-00560 Helsinki
☎ +358 9 2510 400
🖷 +358 9 2510 4010
✉ info@comsol.fi
💻 www.comsol.fi

France

COMSOL France
WTC, 5 pl. Robert Schuman
F-38000 Grenoble
☎ +33 (0)4 76 46 49 01
🖷 +33 (0)4 76 46 07 42
✉ info@comsol.fr
💻 www.comsol.fr

Germany

COMSOL Multiphysics GmbH
Berliner Str. 4
D-37073 Göttingen
☎ +49-551-99721-0
🖷 +49-551-99721-29
✉ info@comsol.de
💻 www.comsol.de

India

COMSOL Multiphysics Pvt. Ltd.
Esquire Centre,
C-Block, 3rd Floor
No. 9, M. G. Road
Bangalore 560001
Karnataka
☎ +91-80-4092-3859
🖷 +91-80-4092-3856
✉ info@comsol.co.in
💻 www.comsol.co.in

Italy

COMSOL S.r.l.
Via Vittorio Emanuele II, 22
25122 Brescia
☎ +39-030-3793800
🖷 +39-030-3793899
✉ info.it@comsol.com
💻 www.comsol.it

Norway

COMSOL AS
Postboks 5673 Sluppen
Søndre gate 7
NO-7485 Trondheim
☎ +47 73 84 24 00
🖷 +47 73 84 24 01
✉ info@comsol.no
💻 www.comsol.no

Sweden

COMSOL AB
Tegnérgatan 23
SE-111 40 Stockholm
☎ +46 8 412 95 00
🖷 +46 8 412 95 10
✉ info@comsol.se
💻 www.comsol.se

Switzerland

COMSOL Multiphysics GmbH
Technoparkstrasse 1
CH-8005 Zürich
☎ +41 (0)44 445 2140
🖷 +41 (0)44 445 2141
✉ info@ch.comsol.com
💻 www.ch.comsol.com

United Kingdom

COMSOL Ltd.
Broers Building
21 J J Thomson Avenue
Cambridge CB3 0FA
☎ +44-(0)-1223 451580
🖷 +44-(0)-1223 367361
✉ info.uk@comsol.com
💻 www.uk.comsol.com

United States

COMSOL, Inc.
1 New England Executive Park
Suite 350
Burlington, MA 01803
☎ +1-781-273-3322
🖷 +1-781-273-6603

COMSOL, Inc.
10850 Wilshire Boulevard
Suite 800
Los Angeles, CA 90024
☎ +1-310-441-4800
🖷 +1-310-441-0868

COMSOL, Inc.
744 Cowper Street
Palo Alto, CA 94301
☎ +1-650-324-9935
🖷 +1-650-324-9936
✉ info@comsol.com
💻 www.comsol.com

For a complete list of international representatives, visit
💻 www.comsol.com/contact

Home Page

💻 www.comsol.com

COMSOL User Forums

💻 www.comsol.com/community/forums

COMSOL Multiphysics Reference Guide

Version: October 2011 COMSOL 4.2a

Part No. CM020005

Contents

Chapter 1: Introduction

Chapter 2: Geometry Modeling Features

Features for Geometric Operations 72

Features for Virtual Operations 98

Chapter 3: Meshing Features

Chapter 4: Definitions

Pairs **201**

Probes **205**

Selections **212**

Views **220**

Chapter 5: Solver Features

The Study Nodes **232**

Study Types 233

Solver Configurations 262

Solution Utility Features **333**

Job Configurations **336**

Chapter 6: Results Analysis and Plots

Chapter 7: The Finite Element Method

Chapter 8: Stabilization Techniques

Stabilization Techniques 500

Chapter 9: Advanced Solver Topics

Solver Algorithms 508

Linear System Solvers 521

Preconditioners for Iterative Solvers 528

Chapter 10: The COMSOL File Formats

1

Introduction

Welcome to COMSOL Multiphysics®! The *COMSOL Multiphysics Reference Guide* is a supplement to the *COMSOL Multiphysics User's Guide* and has a variety of in-depth reference information that is useful while using COMSOL Multiphysics.

In this section:

- Navigating the Documentation
- Overview of the Reference Guide

Navigating the Documentation

In this section:

- Where Do I Access the Documentation and Model Library?
- Checking for Software Updates
- Typographical Conventions

Where Do I Access the Documentation and Model Library?

A number of Internet resources provide more information about COMSOL Multiphysics, including licensing and technical information. The electronic documentation, Dynamic Help, and the Model Library are all accessed through the COMSOL Desktop.

❗ *Important*	If you are working directly from a PDF on your computer, the blue links do not work to open a model or documentation referenced in a different user's guide. However, if you are using the online help in COMSOL Multiphysics, these links work to other modules, model examples, and documentation sets.

THE DOCUMENTATION

The *COMSOL Multiphysics User's Guide* and *COMSOL Multiphysics Reference Guide* describe all interfaces and functionality included with the basic COMSOL license. These guides also have instructions about how to use COMSOL Multiphysics, and how to access the documentation electronically through the COMSOL Multiphysics help desk.

To locate and search all the documentation, in COMSOL Multiphysics:

- Click the buttons on the toolbar, or
- Select **Help>Documentation** (?) or **Help>Dynamic Help** (?) from the main menu

and then either enter a search term or look under a specific module in the documentation tree.

THE MODEL LIBRARY

Each model comes with a theoretical background and step-by-step instructions to create the model. The models are available in COMSOL as MPH-files that you can open for further investigation. Use both the step-by-step instructions and the actual models as a template for your own modeling and applications. SI units are used to describe the relevant properties, parameters, and dimensions in most examples, but other unit systems are available.

To open the Model Library, select **View>Model Library** ( ) from the main menu, and then search by model name or browse under a Module folder name. If you also want to review the documentation explaining how to build a model, select the model and click **Model PDF** or the **Dynamic Help** button ( ) to reach the PDF or HTML version, respectively. Alternatively, select **Help>Documentation** in COMSOL and search by name or browse by Module.

If you have feedback or suggestions for additional models for the library (including those developed by you), feel free to contact us at info@comsol.com.

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 WEB SITES

Main Corporate web site	www.comsol.com
Worldwide contact information	www.comsol.com/contact
Technical Support main page	www.comsol.com/support
Support Knowledge Base	www.comsol.com/support/knowledgebase
Product updates	www.comsol.com/support/updates
COMSOL User Community	www.comsol.com/community

Checking for Software Updates

COMSOL provides software updates that improve the software and correct issues found. To check is an update is available, choose **Check For Updates** from the **Help** menu. 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 the **Browse Update** button 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 is up-to-date.

Typographical Conventions

All COMSOL user guides use 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:

- Click text highlighted in blue to go to other information in the PDF. When you are using the online help desk in COMSOL Multiphysics, these links also work to other Modules, model examples, and documentation sets.
- 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.
- The names of other items on the COMSOL Desktop that do not have direct labels contain a leading uppercase letter. For instance, the Main toolbar is often referred to— the horizontal bar containing several icons that are displayed on top of the user interface. However, nowhere on the COMSOL Desktop, nor the toolbar itself, includes the word “main.”
- The forward arrow symbol **>** are instructions to select a series of menu items in a specific order. For example, **Options>Results** is equivalent to: From the **Options** menu, select **Results**.
- A `Code` (monospace) font indicates you are to make a keyboard entry in the user 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. An italic *`Code`* (monospace) font indicates user inputs and parts of names that can vary or be defined by the user.

- 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.

THE DIFFERENCE BETWEEN NODES, BUTTONS, AND ICONS

Node: A node is located in the **Model Builder** and has an icon image to the left of it. Right-click a node to open a context menu and to perform actions.

Button: Click a button to perform an action. Usually located on a toolbar (the main toolbar or the Graphics toolbar, for example), or in the upper-right corner of a **Settings** window.

Icon: An icon is an image that displays on a window (for example, the **Model Wizard** or **Model Library**) or displays in a context menu when a node is right-clicked. Sometimes selecting an item with an icon from a node's context menu adds a node with the same image and name, sometimes it simply performs the action indicated (for example, **Delete**, **Enable**, or **Disable**).

KEY TO THE GRAPHICS

Throughout the documentation, additional icons are used to help navigate the information. These categories are used to draw your attention to the information based on the level of importance, although it is always recommended that you read these text boxes.

Caution

A Caution icon ⚠ is used to indicate 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, for example:

⚠ *Caution*	This may limit the type of boundary conditions that you can set on the eliminated species. The species selection must be carefully done.

Important

An Important icon is used to indicate 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, for example:

Important	Do not select any domains that do not conduct current, for example, the gas channels in a fuel cell.

Note

A Note icon is used to indicate that the information may be of use to the user. It is recommended that the user read the text, for example:,

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

Tip

A Tip icon is used to provide information, reminders, short cuts, suggestions of how to improve model design, and other information that may or may not be useful to the user, for example:

Tip	It can be more accurate and efficient to use several simple models instead of a single, complex one.

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 document, the text indicates this, for example:

See Also	• Theory for the Single-Phase Flow Interfaces • The Laminar Flow Interface in the *COMSOL Multiphysics User’s Guide*

Model

The Model icon is used in the documentation as well as in COMSOL Multiphysics from the **View>Model Library** menu. If you are working online, 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. These examples occur in the *COMSOL Multiphysics User's Guide*. The Model Library path describes how to find the actual model in COMSOL Multiphysics.

Model	• Acoustics of a Muffler: Model Library path **COMSOL_Multiphysics>Acoustics>automotive_muffler** • If you have the RF Module, see Radar Cross Section: Model Library path **RF_Module>Tutorial_Models>radar_cross_section**.

Space Dimension Icons

Another set of icons are also used in the Model Builder—the model 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 an interface, feature node, or theory section, which are based on space dimension.

The following tables are examples of these space dimension icons.

3D	3D models often require more computer power, memory, and time to solve. The extra time spent on simplifying a model is time well spent when solving it.

2D	Remember that modeling in 2D usually represents some 3D geometry under the assumption that nothing changes in the third dimension.

Overview of the Reference Guide

The *COMSOL Multiphysics Reference Guide* is a supplement to the *COMSOL Multiphysics User's Guide* and has a variety of information that is useful while using COMSOL Multiphysics.

Tip

As detailed in the section Where Do I Access the Documentation and Model Library? this information is also searchable from the COMSOL Multiphysics software **Help** menu.

TABLE OF CONTENTS AND INDEX

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

GEOMETRY

The Geometry Modeling Features chapter describes the geometric primitives, tools, and operations available for modeling the geometry in COMSOL Multiphysics. It includes topics such as The Model Nodes, Building a Geometry Sequence, The Geometry Toolbar, Geometry Features, and Features for Geometric Operations.

MESH

The Meshing Features chapter describes the features available for meshing the geometry and includes topics such as Building Meshing Sequences, Meshing Features, Operation Features, and The Mesh Toolbar.

GLOBAL AND LOCAL DEFINITIONS

The Definitions chapter describes the available functionality on the Global Definitions and Definitions branches. It also includes a description of the Materials nodes. It includes the following topics: Parameters, Variables, Functions, Materials, Coordinate Systems, Selections, and Views.

SOLVER

The Solver Features chapter describes the solver features available in COMSOL Multiphysics. The topics include The Study Nodes, Study Types, Solution Operation Features, Solution Utility Features, and Job Configurations.

RESULTS

The Results Analysis and Plots chapter describes the features for visualization and analysis of the simulation results in COMSOL Multiphysics. The topics are Results, Data Sets, Plot Groups and Plotting, Derived Values and Tables, and Exports.

THE FINITE ELEMENT METHOD

The Finite Element Method chapter contains a theoretical background to the finite element method (FEM) and an overview of the finite element types in COMSOL Multiphysics. Sections also explain how COMSOL Multiphysics forms the system of equations and constraints that it solves and the implications of Dirichlet conditions involving several solution components in a multiphysics model. The chapter also includes a theoretical background to the weak formulation of PDEs and an overview of PDE, energy, and weak form formulations.

STABILIZATION

The Stabilization Techniques chapter describes when spurious oscillations can arise. To restore numerical stability COMSOL Multiphysics provides a set of stabilization techniques. The topics are Numerical Stabilization and Stabilization Techniques.

ADVANCED SOLVER

The Advanced Solver Topics chapter examines some of the solvers and solver algorithms in COMSOL Multiphysics in some detail. The topics are Solver Algorithms, Linear System Solvers, and Preconditioners for Iterative Solvers.

FILE FORMATS

The COMSOL File Formats chapter describes the COMSOL file formats including the COMSOL Multiphysics files in binary format and text format. The topics include File Formats, Data Formats, Binary Data Files and Text Data Files, and Serializable Types.

2

Geometry Modeling Features

This chapter describes the geometric primitives, tools, and operations available for modeling the geometry in COMSOL Multiphysics.

In this chapter:

The Model Nodes

Adding a Model (Geometry)

To add a new model, right-click the root node of the **Model Builder**, and select **Add Model**. This opens the **Model Wizard** window with the heading **Select Space Dimension**. This window also shows up when you launch the COMSOL Desktop and select a 3D, 2D axisymmetric, 2D, 1D axisymmetric, 1D, or 0D geometry. Clicking the **Finish** button (🏁) on the toolbar to close the **Model Wizard** window and create a new, empty model with a geometry sequence only containing a **Form Union** node.

Alternatively, clicking **Next** takes you to the **Add Physics** page, where you can add the physics interfaces you want to use. The current section focuses on building geometries.

See Also

For details about the Add Physics page, see The COMSOL Multiphysics Physics Interfaces in the *COMSOL Multiphysics User's Guide*.

Building a Geometry Sequence

This following sections describe how to work with the geometry sequence:

- Adding and Building Geometry Features
- Editing and Building Geometry Feature Nodes
- Automatic Building of Geometry Features
- Errors and Warnings
- Deleting, Disabling, and Enabling Geometry Features

See Also

- The Geometry Node and Its Settings
- The Geometry Toolbar
- The Measurements Page
- The Export Geometry to File
- Geometry Modeling and CAD Tools in the *COMSOL Multiphysics User's Guide*
- Geometry API in the *COMSOL Java API Reference Guide*

Adding and Building Geometry Features

Add and build geometry features in two ways:

- Right-click a **Geometry** node in the Model Tree and then select one of the available features. Enter the feature's properties in the **Settings** window. In numerical fields, enter expressions that contain parameters defined in **Parameters** under **Global Definitions** in the Model Tree. Click the **Build Selected** button ( ) in the **Settings** form to see the geometry objects that result. All features can be added using this method.

- Use the buttons in the Geometry toolbar. This method is available for some drawing primitives in 2D, and for some operations and conversions acting on geometry objects.

See Also

- The Geometry Toolbar

ADDING A FEATURE NODE AT AN ARBITRARY POSITION

When you add a feature node, it is inserted in the geometry sequence after the *current feature node*. To indicate the current feature node, it appears with a quadratic frame around its icon. To add a feature node after an existing feature 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 feature node.

To add a feature node before an existing feature node, first select the existing node, right-click the existing node and select **Build Preceding** ( ). Then add the feature node.

When you have added a feature node, it automatically becomes current, but COMSOL Multiphysics does not build it automatically.

Editing and Building Geometry Feature Nodes

To edit a feature node, select it in the tree, and make changes in the **Settings** window that appears. Features nodes that you have edited display with an asterisk (*) at the upper-right corner of their icons in the **Model Builder** window. Feature nodes that depend on the edited node display with a yellow triangle at the lower-right corner of the feature 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 feature. 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 feature nodes (if needed) from the first up to the selected feature node.
- Click the **Build All** button ( ) in the **Settings** window, or right-click the **Geometry** node in the tree and select **Build All**. This builds all feature nodes in the sequence above the **Finalize** feature (if needed).

The **Graphics** window shows the resulting geometry objects from the feature nodes that have been built. The results of subsequent features is not visible. The last built feature node becomes the *current feature node* and appears with a quadratic frame around the node's icon. The frame is green if the feature node and all preceding feature nodes are built; that is, if the objects in the **Graphics** window are up to date. The frame is yellow if the feature node or some preceding feature node has been edited since the feature was built and thus needs to be rebuilt.

Automatic Building of Geometry Features

In some situations, the software automatically builds geometry features. For example, when you:

- Add a feature node, the software builds the current feature node (and all its preceding feature nodes) before showing the **Settings** window for the new feature node.
- Generate a mesh or solve the model, the software builds the finalized geometry; that is, it builds all feature nodes if needed.
- Select a feature node that uses the finalized geometry, like physics interface features and mesh features, the software builds the finalized geometry provided that only the last two feature nodes have been edited.
- Enter a work plane, the software builds all feature nodes preceding the work plane node.
- Open a CAD Defeaturing tools page, the software builds the current feature node (and all its preceding feature nodes).

Errors and Warnings

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

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

Deleting, Disabling, and Enabling Geometry Features

- To delete a feature node, right-click it and select **Delete** (✖).
- To delete a geometry sequence, right-click the **Geometry** node in the **Model Builder** and select **Delete Sequence** ( ). Click **Yes** or **No** to confirm the deletion. You cannot use the **Undo** command.
- To delete entities, in the **Model Builder**, right-click **Geometry** and select **Delete Entities** ( ).Under **Input**, select a **Geometric entity level** to delete—**Object**, **Domain**, **Boundary**, **Edge**, or **Point**. Select the geometry in the **Graphics** window. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required
- To disable a feature node, right-click it and select **Disable** (⊘). The disabled feature node does not affect the finalized geometry. This is indicated with a gray icon and **[ ]** (brackets) around the node name in the model tree.
- To enable a disabled feature node, right-click it and select **Enable** (●).

The Geometry Node and Its Settings

Under a **Geometry** node ( ) you find the sequence of feature nodes that define the geometry for the model. The **Geometry** node also contains some general settings for the geometry such as the length unit. To open the **Settings** window for a geometry, click the **Geometry** node in the Model Builder.

UNITS

The following properties appear under **Units**:

Length Unit

From the **Length unit** list select the length unit to use in fields for lengths and for visualization of the geometry. Override the unit by 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, meter. If you change the unit, the program 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 below).

Note

For information about available length units and prefixes, see Specifying Model Equation Settings in the *COMSOL Multiphysics User's Guide*.

Angular Unit

From the **Angular unit** list choose to use radians or degrees as the angular unit to use in fields for angles. Override the unit by entering, for example, `0.3[rad]`. The program assumes that numeric inputs and outputs of trigonometric functions are always in radians, though. 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 below).

Scale Values When Changing 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 size. The default setting is to not scale the values when changing units; the software then interprets the values for the geometric dimensions using the new units for length and angle. The values themselves do not change.

ADVANCED

The following properties appear under **Advanced**:

Geometry Representation (3D Only)

The **Geometry representation** list controls which kernel (geometric modeler) that COMSOL Multiphysics uses to represent and operate on the geometry objects: COMSOL Multiphysics' own kernel or the CAD Import Module's kernel (Parasolid).

- If you choose **COMSOL kernel**, all objects are represented using COMSOL's kernel.
- If you choose **CAD Import Module kernel**, all features that support this use the CAD Import Module's kernel. For example, the Work Plane, Extrude, and Revolve features currently do not support this kernel. This alternative requires that you have a license for the CAD Import Module.

The default geometry representation is controlled by the preference setting **Geometry>Default Geometry representation**.

If the chosen geometry representation cannot be used—for example, because a feature requires the CAD Import Module's kernel or because no CAD license is available—the behavior is controlled by the preference setting **Geometry>Switch kernel automatically**. If automatic switching is enabled, the software automatically switches to the other geometry kernel when needed. If automatic switching is disabled, an error message is generated if the selected kernel cannot be used.

When you change the **Geometry representation** setting, all feature nodes that support the CAD Import Module's kernel are marked as edited with an asterisk (*) in the upper-right corner the feature node's icon. To rebuild the geometry using the new kernel, click the **Build All** button ( ) or the **Build Selected** button ( ).

Default Relative Repair Tolerance

This is the default value that is used when you add a new 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.

The Geometry Toolbar

The Geometry toolbar is a part of the Main toolbar and has buttons to perform geometric operations. The Mesh toolbar is available when a geometry sequence node is selected in the **Model Builder**. This means that the Geometry toolbar disappears if you, for example, select a geometry node in the **Model Builder**. Use the **Toolbar** menu button ( ) in the Main toolbar to make the Geometry toolbar for a specific geometry node available; click the associated menu arrow and choose the menu item corresponding to the geometry 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.

To use the buttons on the toolbar, first select geometry objects in the **Graphics** window. When you click one of the buttons, COMSOL Multiphysics performs the associated operation on the selected objects and creates they resulting objects. In addition, the software adds a feature node for the operation to the geometry sequence. If you want to modify the operation, you can edit and rebuild this feature node. You do not need to confirm the selections in the **Graphics** window when working with the Geometry toolbar; it is sufficient to highlight them, a state that is indicated by the color red.

In 2D there are also buttons for drawing geometric primitives. To use these, first click the button and then draw the object in the **Graphics** window using the mouse. Then the primitive geometry object appears in the graphics. In addition, the software adds a feature node for the primitive to the geometry sequence. If you want to modify the primitive, you can edit and rebuild this feature.

The Geometry toolbar contains the following parts:

DRAW OPERATIONS (1D AND 2D ONLY)

- To draw a point, first click the **Draw Point** button (●), then click in the **Graphics** window.
- To draw an interval in 1D, first click the **Draw Interval** button ( ), then click the start and end points in the **Graphics** window.
- To draw a polygon consisting of line segments or Bézier curves in 2D, first click one of the buttons **Draw Line** ( ), **Draw Quadratic** ( ), or **Draw Cubic** ( ). 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

Draw Line ( ), **Draw Quadratic**( ), or **Draw Cubic** ( ) 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** feature 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.

- To draw a square, click **Draw Square** ( ) or **Draw 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** feature node appears in the geometry sequence.
- To draw a rectangle, click **Draw Rectangle** ( ) or **Draw 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** feature node appears in the geometry sequence.
- To draw a circle, click **Draw Circle (center)** ( ) or **Draw Circle** ( ). 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** feature node appears in the geometry sequence.
- To draw an ellipse, click **Draw Ellipse (center)** ( ) or **Draw Ellipse** ( ). 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** feature node appears in the geometry sequence.

By default, the mouse pointer snaps to the grid points. To disable snapping, click the **Snap Coordinates** button ( ).

BOOLEAN OPERATIONS

You select the objects you want to compose in the **Graphics** window, and then click one of the buttons **Union** ( ), **Intersection** ( ), or **Difference** ( ). **Intersection** and **Difference** only apply to solid objects. The **Difference** button takes the selected object with largest volume (area, length) and subtracts the others.

CONVERSIONS

You select the objects you want to convert in the **Graphics** window, and then click one of the buttons **Convert to Solid** ( ), **Convert to Surface** ( ) (only 3D), **Convert to**

Curve ( ) (not 1D), **Convert to Point** ( ), or **Convert to COMSOL** ( ) (only if you have license for the CAD Import Module).

OTHER OPERATIONS

- To edit a 2D object, select the object and click the **Edit Object** button ( ).
- To split one or several objects into their entities, select them in the **Graphics** window, and then click the **Split** button ( ).
- 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 their nodes disappear from the sequence. If you delete other objects or if you delete geometric entities a **Delete Entities** node ( ) appears in the sequence.

MEASUREMENTS

- To measure the volume, area, or length of a selected domain, face, or edge, respectively, click the **Measure** button ( ). The result appears 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 an object.

The Measurements Page

The **Measurements** page is a tool to measure geometry objects and entities. You access it by right-clicking the Geometry node and selecting **Measurements** (📏) from the context menu.

GEOMETRY TYPE

Select to measure geometry objects 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. Under **Measurements** you find information about these objects or entities. For objects, this section contains their 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, their distance is shown.

Insert Sequence from File

To insert a geometry sequence from an MPH-file, right-click the **Geometry** node and select **Insert Sequence from File** ( ) from the context menu. 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 features are inserted into the geometry sequence after the current feature 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 features in the selected sequence are inserted into the geometry sequence after the current feature 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 model the geometry sequence is located.

Note	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.

The Export Geometry to File

Geometry objects can be exported to a COMSOL Multiphysics binary file (`mphbin`) or text file (`mphtxt`). If you have a license for the CAD Import Module, geometry objects can also be exported to a Parasolid file (`x_t`, `x_b`).

2D

2D objects can also be exported to a DXF file.

To open the **Export Geometry** page, right-click the **Geometry** node and select **Export to File** ( ) from the menu. Select an option—**Export selected objects** or **Export entire finalized geometry**. If **Export selected objects** is chosen, select the objects to export from the **Input** list or in the **Graphics** window.

Tip

To export a geometry to use in an earlier version of COMSOL, select the applicable version from the **Compatible with version** list.

Click **Export** and in the **Export Geometry to File** window and navigate to the location to export the file. Enter a **File name** and if required, select a different file type from the **Save as type** list and then click **Save**.

Geometry Features

COMSOL Multiphysics has a number of features to create geometric primitives and to operate on them, for example, to add fillets or create a composite solid object from other solid objects. There is also a feature for finalizing the geometry.

In this section:

- Features for Creating Geometric Primitives
- Features for Geometric Operations
- The Finalize Feature Node
- Features for Virtual Operations
- Backward Compatibility

Features for Creating Geometric Primitives

Table 2-1 is an overview of the features to create 3D geometric primitives:

TABLE 2-1: 3D GEOMETRIC PRIMITIVES

NAME	ICON	DESCRIPTION
Bézier Polygon		Chain of connected line segments, quadratic or cubic curves
Block		Right-angled parallel (box)
Cone		Right circular cone or cone frustum
Cylinder		Right circular cylinder
Eccentric Cone		Oblique cone or cone frustum with elliptic base
Ellipsoid		Ellipsoid
Helix		Helix solid, surface, or curve
Hexahedron		Hexahedron bounded by bilinear faces
Interpolation Curve		A curve defined by a sequence of point coordinates
Parametric Curve	f(s)	A curve defined by its coordinates as function of a parameter

TABLE 2-1: 3D GEOMETRIC PRIMITIVES

NAME	ICON	DESCRIPTION
Parametric Surface	▱	A surface defined by its coordinates as functions of two parameters
Point	●	One or several points
Polygon	▱	Chain of connected line segments
Pyramid	⊠	Rectangular pyramid
Sphere	●	Sphere or ball
Tetrahedron	▲	Tetrahedron
Torus	◎	Torus

Table 2-2 is an overview of the features to create 2D geometric primitives.

TABLE 2-2: 2D GEOMETRIC PRIMITIVES

NAME	ICON	DESCRIPTION
Bézier Polygon	∿	Chain of connected line segments, quadratic curves, or cubic curves
Circle	○	Circle or disc
Ellipse	⬭	Ellipse
Interpolation Curve	∿	A curve defined by a sequence of point coordinates
Parametric Curve	f(s)	A curve defined by its coordinates as function of a parameter
Point	●	One or several points
Polygon	▱	Chain of connected line segments
Rectangle	▭	Rectangle
Square	□	Square

Table 2-3 is an overview of the features to create 1D geometric primitives.

TABLE 2-3: 1D GEOMETRIC PRIMITIVES

NAME	ICON	DESCRIPTION
Interval		One interval, or a chain of connected intervals
Point	●	One or several points

Features for Geometric Operations

The **Import** feature imports geometry objects from a file or from another geometry. The following tables list the features that create new geometric objects from existing ones.

TABLE 2-4: WORK PLANE RELATED FEATURES (3D ONLY)

NAME	ICON	DESCRIPTION
Work Plane		Create a work plane for drawing 2D objects that are embedded in 3D
Extrude		Extrude planar objects from a work plane into 3D
Revolve		Revolve planar objects from a work plane into 3D
Sweep		To sweep a face along a spine curve

TABLE 2-5: BOOLEAN OPERATIONS

NAME	ICON	DESCRIPTION
Compose		Compose solids using a set formula
Difference		Subtract solids from solids
Intersection		Intersect solids
Union		Unite geometry objects

TABLE 2-6: LINEAR TRANSFORMATIONS

NAME	ICON	DESCRIPTION
Array		Rectangular or linear array of geometry objects
Copy		Make displaced copies of geometry objects

TABLE 2-6: LINEAR TRANSFORMATIONS

NAME	ICON	DESCRIPTION
Mirror		Reflect objects in a plane (in 3D), a line (in 2D), or a point (in 1D)
Move		Translate geometry objects
Rotate		Rotate geometry objects about a center point
Scale		Scale a geometric objects about a center point

TABLE 2-7: OBJECT TYPE CONVERSIONS

NAME	ICON	DESCRIPTION
Convert to Solid		Unite and convert objects to a single solid object
Convert to Surface		Unite and convert 3D objects to a single surface object
Convert to Curve		Unite and convert 2D or 3D objects to a single curve object
Convert to Point		Unite and convert objects to a single point object

TABLE 2-8: OTHER OPERATIONS

NAME	ICON	DESCRIPTION
Chamfer		Chamfer corners in 2D geometry objects
Delete Entities		Delete entities (domains, boundaries, edges, or points) from objects, or delete entire geometry objects
Edit Object		Edit the edges and vertices in 2D geometry objects
Fillet		Fillet corners in 2D geometry objects
Object Selection		Create a selection of geometric entities for a geometry object
Split		Split geometry objects into their constituent entities
Tangent		Tangent to an edge in 2D

The Finalize Feature Node

The **Finalize** node (🏁) ends each geometry sequence in 1D and 2D. In 3D, it is possible to add virtual operation nodes after the **Finalize** 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 **Finalize** feature. The output object of the last feature of a sequence is called the *finalized geometry*, which is used for meshing and analysis.

FINALIZE

The default finalization method is to form a union of all geometry objects. Select **Form an assembly** from the **Finalization method** list if you do not want the geometry objects to be united. The program then forms the finalized 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. Select the **Create pairs** check box to generate pairs corresponding to the parts of the assembly that touch each other. From the **Pair type** list, select **Identity pair** (the default) to generate identity pairs or select **Contact pair** to generate contact pairs. The contact pairs are only useful for contact modeling in structural mechanics, which requires a license for the Structural Mechanics Module or the MEMS Module.

Adjust the value in the **Relative repair tolerance** field if you experience problems with the finalization operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of all geometry objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

Features for Virtual Operations

Table 2-9 is an overview of the features for virtual operations in 3D:

TABLE 2-9: VIRTUAL OPERATION FEATURES (ONLY 3D)

NAME	ICON	DESCRIPTION
Ignore Vertices	✕	Virtually remove isolated vertices or vertices adjacent to two edges only
Ignore Edges	✕	Virtually remove isolated edges or edges adjacent to two faces only
Form Composite Edges	+ +	Form virtual composite edges from sets of connected edges by ignoring the vertices between the edges in each set

TABLE 2-9: VIRTUAL OPERATION FEATURES (ONLY 3D)

NAME	ICON	DESCRIPTION
Form Composite Faces	⊞	Form virtual composite faces from sets of connected faces by ignoring the edges between the faces in each set
Collapse Edges	•—•	Virtually collapse each edge into a vertex by merging its adjacent vertices
Merge Vertices	•—•	Virtually merge one adjacent vertex of an edge with the other adjacent vertex

Backward Compatibility

If you open a model created in the 3.5a version of COMSOL Multiphysics one **Composite Object** node appears for each nonprimitive geometry object in the model. See Composite Object for more information.

Features for Geometric Primitives

The following sections describe the feature nodes for creating geometric primitives:

- Bézier Polygon
- Block
- Circle
- Cone
- Cylinder
- Eccentric Cone
- Ellipse
- Ellipsoid
- Helix
- Hexahedron
- Interpolation Curve
- Interval
- Parametric Curve
- Parametric Surface
- Point
- Polygon
- Pyramid
- Rectangle
- Sphere
- Square
- Tetrahedron
- Torus

Bézier Polygon

A Bézier polygon consists of a sequence of connected line segments, quadratic Bézier curves (for example circular arcs), and cubic Bézier curves. To create a Bézier polygon, right-click a 2D **Geometry** node and select **Bézier Polygon** () or right-click a 3D

Geometry node and select **More Primitives>Bézier Polygon**. Then enter the properties of the Bézier polygon.

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.

POLYGON SEGMENTS

Define the Bézier polygon by adding curve segments to the list of segments. Choose from linear segments, quadratic segments, and 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 (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.

Block

To create a block (box), right-click a 3D **Geometry** node and select **Block** ( ). 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 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 **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 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.

Circle

To create a circle or disk, right-click a 2D geometry and select **Circle**. Then enter the properties of the circle using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the circle is a solid object or a curve object.

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). From the **Base** list, choose **Center** if the circle 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 position in the **Rotation** field.

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.

Cone

To create a right circular cone or cone frustum, right-click a 3D **Geometry** and select **Cone** (▶). Then 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 surface object.

SIZE AND SHAPE

Define the size and shape of the cone in the **Radius**, **Height**, and **Semi-angle** fields. The semi-angle is the angle a cone makes with the vertical axis (in degrees). The semi-angle can take values between 0 degrees (which makes the cone into a cylinder) and 45 degrees (for a cone with a sharp apex).

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 **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 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.

Cylinder

To create a right circular cylinder, right-click a 3D **Geometry** node and select **Cylinder** ( ). 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 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 **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 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.

Eccentric Cone

To create an eccentric (oblique) cone or cone frustum with elliptic base, right-click a 3D **Geometry** node and select **More Primitives>Eccentric Cone** ( ). 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 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 **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 (in degrees) about the axis in the **Rotation** field. When this angle is zero, 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.

Ellipse

To create an ellipse, right-click a 2D geometry and select **Ellipse** ( ). Then enter the properties of the ellipse using the following sections:

GENERAL

From the **Type** list, select **Solid** or **Curve** to specify if the ellipse is a solid object or a curve object.

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). 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.

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.

Ellipsoid

To create an ellipsoid, right-click a 3D **Geometry** node and select **More Primitives>Ellipsoid** ( ). 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 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 **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 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.

Helix

To create a helix (coil), right-click a 3D **Geometry** node and select **More Primitives>Helix** ( ). Then enter the properties of the helix using the following sections:

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** field contains the radius from the center of the helix (default value: 1).
- The **Minor radius** field contains the radius of the cross section (default value: 0.1). 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 determines the axial distance between similar positions on two consecutive turns of the helix (default value: 0.3).
- The **Radial pitch** field determines the radial distance between similar positions on two consecutive turns of the helix (default value: 0, which means that each turn has the same radius).

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. Select **Right handed** or **Left handed** from the **Chirality** list.

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

Define the axis for the helix.

- Select **Cartesian** from the **Axis type** list 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** from the **Axis type** list 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 giving an angle in the **Rotation** field. The default value is 0.

ADVANCED SETTINGS

By default, twist compensation is active and prevents the twisting that would otherwise occur due to nonzero torsion for curves that do no belong to a fixed plane. Clear the **Twist compensation** check box to turn off this compensation.

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.

Hexahedron

To create a hexahedron bounded by bilinear faces, right-click a 3D **Geometry** node and select **Hexahedron** (). 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 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.

Interpolation Curve

An interpolation curve consists of a curve that interpolates or approximates a sequence of points. To create an interpolation curve, right-click a 2D **Geometry** node and select **Interpolation Curve** ( ) or right-click a 3D **Geometry** node and select **More Primitives>Interpolation Curve**. 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 directly in the **Settings** window. Select **File** from the **Data source** list to read point coordinate data from a text file.

If **Data source** is **File**, specify the file name 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**.

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.

Interval

To create one or several intervals, right-click a 1D geometry and select **Interval** ( ). Then enter the properties of the interval using the following section:

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 entities (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.

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·cos(s), s·sin(s)) for a parameter s that runs from 0 to 10π defines a spiral in 2D. To create a parametric curve, right-click a 2D geometry and select **Parametric Curve** ( ), or a 3D geometry and select **More Primitives>Parametric Curve** ( ). 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** and **Maximum** fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each space coordinate in the **x**, **y** (**r** and **z** in 2D axial symmetry), 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.

Note

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), 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), 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 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. 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** (**r** and **z** in 2D axial symmetry), and (in 3D) **z** 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.

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, right-click a 3D geometry and select **More Primitives>Parametric Surface** ( ). 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** and **Maximum** fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each space 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 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. 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 selection that includes the parametric surface's edges) but do not appear as separate selection nodes in the Model Tree.

Point

To create one or several points, right-click a 1D, 2D, or 3D geometry and select **Point** (●) (in 3D: **More Primitives>Point**). Then enter the properties of the point using the following section:

POINT

Define the position of the point by entering its coordinates in fields labeled **x**, (2D and 3D only) **y**, and (3D only) **z**; **r** and **z** in 2D axial symmetry. 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 do not appear as a separate selection node in the Model Tree.

Polygon

A polygon consists of a sequence of connected line segments. To create a polygon, right-click a 2D **Geometry** node and select **Polygon** ( ) or right-click a 3D **Geometry** node and select **More Primitives>Polygon**. 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 software automatically adds a line segment if needed to close the polygon.

COORDINATES

Define the coordinates of the vertices in the fields **x**, **y**, and (3D only) **z**; **r** and **z** in 2D axial symmetry. Each field can contain a list of numbers or expressions containing parameters, separated with commas or spaces.

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.

Pyramid

To create a rectangular pyramid or pyramid frustum, right-click a 3D **Geometry** node and select **More Primitives>Pyramid** . 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 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 determines the side lengths of the bottom rectangle. The **Height** fields determines the height of the pyramid frustum. The **Ratio** field controls the ratio of the perimeters of the top and bottom rectangles. To get an oblique pyramid, use the **Top displacement** fields 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 **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 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.

Rectangle

To create a rectangle, right-click a 2D geometry and select **Rectangle**. Then enter the properties of the rectangle using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the rectangle is a solid object or a curve object.

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). 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 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.

Sphere

To create a sphere or ball, right-click a 3D **Geometry** node and select **Sphere** ●. 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 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 **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 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 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 sphere consists of. These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the Model Tree.

Square

To create a square, right-click a 2D geometry and select **Square** . Then enter the properties of the square using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the square is a solid object or a curve object.

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). 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.

Tetrahedron

To create a tetrahedron, right-click a 3D **Geometry** node and select **More Primitives>Tetrahedron**.

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the tetrahedron is a solid object or a 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.

Torus

To create a torus, right-click a 3D **Geometry** node and select **More Primitives>Torus**.

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the torus is a solid object or a surface object.

SIZE AND SHAPE

Define the size and shape of the torus in the **Major radius**, **Minor radius**, and **Revolution angle** fields. The **Major radius** field controls the radius of the directrix circle. The **Minor radius** field controls the radius of generatrix circle (the cross section). To get less than a full revolution (360 degrees), use the **Revolution angle** field (angle unit: degrees).

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 **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.

Features for Geometric Operations

These topics are included in this section for performing geometric operations:

- Array
- Chamfer
- Compose
- Convert to Curve
- Convert to Point
- Convert to Solid
- Convert to Surface
- Copy
- Deformed Configuration
- Delete Entities
- Difference
- Edit Object
- Extrude
- Fillet
- Import
- Intersection
- Mirror
- Move
- Object Selection
- Revolve
- Rotate
- Scale
- Split
- Sweep
- Tangent
- Union
- Work Plane

Array

To create an array of geometry objects, right-click a geometry and select **Transforms>Array** ( ). 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.

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**, **y**, and **z** fields; **r** and **z** in 2D axial symmetry.

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.

Chamfer

To chamfer corners in 2D geometry objects, right-click a geometry and select **Chamfer** ( ). Then enter the properties of the chamfer operation using the following sections:

POINT SELECTION

Select the vertices that you want to chamfer in the Graphics window. They then appear in the **Vertices to chamfer** list.

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.

Compose

To create a composite geometry object from solids using Boolean operations given in a set formula, right-click a geometry and select **Boolean Operations>Compose** (). Then enter the properties of the compose operation using the following section:

COMPOSE

Select the solid objects that you want to compose in the Graphics window. The solid objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected solid geometry objects for further geometric operations.

In **Set formula**, enter a set formula involving the names of the selected geometry objects, for example `r1+c1*(c2-r2)`. Use the binary operations `+`, `*`, and `-` for union, intersection, and set difference, respectively. The precedence of the operators `+` and `-` are the same. The operator `*` has higher precedence. Override the precedence rules using parentheses. When you change the set formula, the **Input objects** selection is automatically updated.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box.

Adjust the **Relative repair tolerance** 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. 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.

Convert to Curve

To unite and convert geometry objects to single a curve object, right-click a geometry and select **Conversions>Convert to Curve** ( ). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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.

Convert to Point

To unite and convert geometry objects to single a point object, right-click a geometry and select **Conversions>Convert to Point** ( ). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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 ia available in all **Selection** lists for points but do not appear as a separate selection node in the Model Tree.

Convert to Solid

To unite and convert geometry objects to single a solid object, right-click a geometry and select **Conversions>Convert to Solid** (). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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.

Convert to Surface

To unite and convert geometry objects to single a surface object, right-click a geometry and select **Conversions>Convert to Surface** (). 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. Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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.

Copy

To create a copy of geometry objects, right-click a geometry and select **Transforms>Copy** ( ).

> Note
>
> There is another way to copy geometry objects. Right-click the geometry object to copy (for example **Rectangle**, **Circle**, **Sphere**, or **Ellipse**) and select **Copy** ( ). Then right-click and select **Paste** {**Rectangle**, **Circle**, **Sphere**, or **Ellipse** for example} ( ) from the menu.

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. 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. 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.

See Also

In the *COMSOL Multiphysics User's Guide:*

- Copying and Pasting Geometry Objects
- Copying, Pasting, and Duplicating Nodes

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**.

See Also

In the *COMSOL Multiphysics User's Guide:*

- Remeshing a Deformed Mesh
- Solution Data Sets

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.

Tip

If you want to delete all objects created by a feature, it is better to right-click the feature, and select **Delete** (✕) or **Disable** (⊘).

INPUT

From the **Geometric entity level** list, choose the level of the entities to delete: **Object**, **Domain**, **Boundary**, **Edge**, 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 list when you have confirmed (locked) the selection in the Graphics window.

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.

See Also

- Working with Geometry in the *COMSOL Multiphysics User's Guide*

Difference

To subtract solids from solids, right-click a geometry and select **Boolean Operations>Difference** ( ). Then enter the properties of the difference operation using the following section:

INPUT

Activate the **Objects to add** list by selecting its **Activate Selection** button ( ), and select the solids that you want to add in the **Graphics** window.

Activate the **Objects to subtract** list by selecting its **Activate Selection** button ( ), and select the solids that you want to subtract in the **Graphics** window.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box.

Adjust the **Relative repair tolerance** 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.

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.

When changing edge control points or vertex coordinates, the solid domains in the object are preserved. However, when changing the object in any other way, for example adding or deleting an edge or a vertex, the object is converted to a curve object.

See Also

- Editing 2D Geometry Objects in the *COMSOL Multiphysics User's Guide*

SELECT OBJECT

Select the **Input Object** to edit in the **Graphics** window.

EDIT EDGES

Click the **Activate Edges** button (⏻). In the **Graphics** window, select and add the **Edge** number to edit (or use the **View>Selection** list method). The parameters for the current edge are displayed in the table for **x (m)**, **y (m)**, 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 or **Delete** to delete the current edge.

Click in the table cells to edit the **x (m)**, **y (m)**, 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.

Under **Start vertex**, click the **Activate Vertices** 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 **Activate Vertices** 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.

Tip

When the **Edit Object** feature is the current feature 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 **Activate Vertices** button (⏻). In the **Graphics** window, select and add the **Vertex** number (node number) to edit (or use the **View>Selection** list method). 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.

Tip

When the **Edit Object** feature is the current feature node, you can visualize the edited object by observing the vertex numbers displayed in the **Graphics** window next to the vertices.

Click **New** to create a new vertex, **Delete** to delete the current vertex, or **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.

Extrude

To extrude objects from a work plane, right-click a geometry or a work plane feature, and select **Extrude** ( ). Then enter the properties of the extrude operation.

GENERAL

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 **Keep input objects** check box to use the selected planar objects in other geometric operations, for instance extrude or revolve. If you extrude several layers, remove the interior boundaries by clearing the **Keep cross-sectional faces** check box.

DISTANCES FROM WORK PLANE

Specify one or several distances in the table. These determine the height above the work plane for the top of each layer. The **Reverse direction** check box reverses the direction of the extrusion.

SCALES

For each layer, specify a length scaling factor for the top of the layer relative to the work plane object.

DISPLACEMENTS

For each layer, specify a displacement vector for the top of the layer in the work plane's coordinate system.

TWIST ANGLES

For each layer, specify a rotation angle for the top of the layer around the work plane's normal vector.

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.

Fillet

To fillet corners in 2D geometry objects, right-click a geometry and select **Fillet** ( ). Then enter the properties of the fillet operation using the following sections:

POINT SELECTION

Select the vertices that you want to fillet in the Graphics window. They then appear in the **Vertices to fillet** list.

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.

Import

To import geometry objects from a file or from another geometry, right-click a geometry and select **Import** (⇦). Then enter the properties of the import feature in the **Import** section of the **Settings** window for the **Import** node.

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.

> *Note*
>
> 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).
>
> If you have license for the AC/DC Module, MEMS Module, or RF Module, you also have the alternatives **ECAD file (GDS/NETEX-G)** and **ECAD file (ODB++)** in 2D and 3D; see the section Importing ECAD Files in the *AC/DC Module User's Guide*, the *MEMS Module User's Guide*, or the *RF Module User's Guide*.

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

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, and select **Do not knit** to do nothing.

If the **Repair imported objects** check box is selected, enter a **Relative repair tolerance**. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the imported objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.

Properties for STL/VRML import

STL and VRML are surface mesh formats. The imported objects are represented using COMSOL's geometry kernel, so you cannot apply CAD defeaturing or repair on them.

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 value **Maximum angle within face**. The angle between neighboring triangles in the same face is less than the value **Maximum face neighbor angle**.

If the check box **Detect planar faces** 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.

Intersection

To create the intersection of solids, right-click a geometry and select **Boolean Operations>Intersection** (⧉). Then enter the properties of the intersection operation using the following section:

INPUT

Select the solid objects that you want to intersect in the Graphics window. The solid objects appear in the **Input objects** list.

Check **Keep input objects** to use the selected geometry objects for further geometric operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box.

Adjust the **Relative repair tolerance** 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.

Mirror

To reflect geometry objects in a plane (3D), a line (2D), or a point (1D), right-click a geometry and select **Transforms>Mirror** ( ). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.

POINT ON PLANE/LINE OF REFLECTION

2D 3D	Specify a point to be fixed during reflection by entering **x**, **y**, and **z**.

NORMAL VECTOR TO PLANE/LINE OF REFLECTION

2D 3D	Specify a vector in the direction to reflect by entering **x**, **y**, and **z**.

POINT OF REFLECTION

1D	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.

Move

To translate geometry objects, right-click a geometry and select **Transforms>Move** (✥). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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. 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.

Object Selection

Use an **Object Selection** node ( ) for selecting all or a few of the geometric entities at a specific geometric entity level for one or more of the nodes above the **Object 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. Right-click the **Object Selection** node and select **Properties** to change its name.

You specify the geometric entities to select in the following section:

OBJECT SELECTION

Under **Entities to select**, select a geometric entity level—**Domain**, **Boundary**, **Edge**, or **Point** in 3D—from the **Geometric entity level** list. Then select the geometric entities to include in the selection using the available selection methods. You can only select geometric entities of the type that is selected from the **Geometric entity level** list.

Revolve

To revolve objects from a work plane, right-click a geometry or a work plane node, and select **Revolve** ( ). Then enter the properties of the revolve operation.

GENERAL

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.

Select the **Keep input objects** check box to use the selected planar objects in other geometric operations, for instance extrude or revolve.

REVOLUTION ANGLES

Select **Angles** in the **Type of specification** list to specify the start and end angle of the revolution. If you select **Full revolution** in the list you get a full revolution. This 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, such faces are kept.

POINT ON THE REVOLUTION AXIS

Specify a point on the revolution axis in the work plane's coordinate system in the **x** and **y** fields.

DIRECTION OF REVOLUTION AXIS

Specify a direction vector for the revolution axis in the work plane's coordinate system in the **x** and **y** 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.

Rotate

To rotate geometry objects, right-click a 3D or 2D geometry and select **Transforms>Rotate** ( ). You can create one or multiple rotated copies with varying rotation angle. Then enter the properties of the rotate operation using the following section:

INPUT

Select the geometry objects that you want to rotate in the Graphics window. The objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.

ROTATION ANGLE

Specify the rotational angle (in degrees) in the **Rotation** field. 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

2D *2D Axi*	Enter the center of the rotation in the **x** and **y** (**r** and **z** in 2D axial symmetry) fields.

POINT ON AXIS OF ROTATION

3D	Enter a point on the rotation axis in the **x**, **y**, and **z** fields.

AXIS OF ROTATION

3D	From the **Axis type** list, 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.

Scale

To scale geometry objects, right-click a geometry and select **Transforms>Scale** ( ). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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.

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. 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.

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, right-click a geometry and 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric 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.

Sweep

To sweep a face along a spine curve, right-click a 3D **Geometry** node and select **Sweep** (). Then enter the properties of the sweep operation using the following sections:

CROSS SECTION

Select the face you want to sweep in the **Graphics** window. Only one face can be selected. The face appears in the **Face to sweep** list. You might have to first click the **Activate Selection** button (⏻).

SPINE CURVE

Select the edge you want to sweep along in the **Graphics** window. Only one edge can be selected. The edge appears in the **Edge to follow** list. You might have to first click the **Activate Selection** button (⏻).

Select the **Reverse direction** check box to sweep in the negative edge direction.

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 torsion of the curve. This makes the edges in the sweep direction locally parallel to the spine curve. In the case of 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} (or 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 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.

Tangent

To create a tangent, that is, a line segment tangent to a specified edge, right-click a 2D **Geometry** node and select **Tangent** (). Then enter the properties of the tangent.

TANGENT

Select the edge you want to tangent in the **Graphics** window. Only one edge may be selected. The edge appears in the **Edge to tangent** list.

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.

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.

Union

To create the union of geometry objects, right-click a geometry and select **Boolean Operations>Union** ( ). 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.

Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box.

Adjust the **Relative repair tolerance** 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.

Work Plane

To create a work plane for defining 2D objects in 3D (for example, to extrude into a 3D object), right-click a 3D **Geometry** node and select **Work Plane** ( ). Then enter the properties defining the location of the work plane.

WORK PLANE

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 between the following alternatives:

- **Quick**—In the **Plane** list, select one of the 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 from the coordinate plane in the field **z-coordinate**, **x-coordinate**, or **y-coordinate**.

- **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.

  In the **Origin** list, you choose the location of the origin of the work plane's coordinate system: **Center of face** or **Bounding box corner**.

  By entering a value in the **Offset in local z direction** list, you can offset the work plane along the normal of the planar face. The work plane's normal is the outward normal of the face in the **Planar face** list.

  To reverse the direction of the *z*-axis of the work plane's coordinate system, select the **Reverse direction of local axis** 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 **Activate Selection** button (⏻) and then selecting an edge in the **Graphics** window. Similarly, activate the **Face adjacent to edge** list by first selecting its **Activate Selection** button (⏻) and then selecting an adjacent face in the **Graphics** window. Also, specify a value in the **Angle between face and work plane** field. 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 direction of local axis** check box, the origin instead is at the end vertex, and the direction of the local *x*-axis is reversed.

- **Vertices**—In each of the lists **First vertex**, **Second vertex**, and **Third vertex**, select a vertex by first selecting the corresponding **Activate Selection** 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 local z direction** field.

  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 direction of local axis** check box.

- **General**—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.

3D Projection

In the **3D projection** list, specify how to visualize 3D objects in the work plane:

- **Coincident 3D geometry**—Only show faces, edges, and points that intersect with the work plane.
- **Entire 3D geometry**—Show the projections of all entities.
- **No visualization**—Do not show any objects outside the work plane.
- Select the **Draw on work plane** in 3D check box to change the behavior when in the **Geometry** node under the **Workplane** node. When this check box is selected, the 3D scene with the workplane displays and the 2D drawing toolbar is activated. Two buttons also become available on the **Graphics** window— the **Align with Workplane** button ( ) to rotate and move the camera to see the workplane from the top down, and the **Work Plane Clipping** button ( ). When toggled ON, use it to cut away all geometries above the workplane and make it easier to draw when objects are overlapping within the workplane.

See Also

- Drawing on the 2D Work Plane in 3D in the *COMSOL Multiphysics User's Guide*

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.

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 **Geometry** node that appears under the **Work Plane** node. To create 2D objects in the work plane, right-click the **Geometry** node and create feature 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**.

Features for Virtual Operations

These topics are included in this section for performing virtual operations:

- Ignore Vertices
- Ignore Edges
- Form Composite Edges
- Form Composite Faces
- Collapse Edges
- Merge Vertices

Ignore Vertices

To remove isolated vertices or vertices adjacent to two edges only from the geometry, right-click a geometry and select **Virtual Operations>Ignore Vertices** ( ). Then enter the properties of the operation using the following sections:

INPUT

Select the vertices that you want to ignore in the Graphics window. They then appear in the **Vertices to ignore** list.

The operation removes the selected vertices that are isolated or that are adjacent to two edges only. However, it does not remove vertices such that it introduces closed or periodic *composite edges*.

Ignore Edges

To remove isolated edges or edges adjacent to two faces only from the geometry, right-click a geometry and select **Virtual Operations>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. They then appear in the **Edges to ignore** list.

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 removes the selected edges that are isolated or that are adjacent to two faces only.

Form Composite Edges

To form virtual composite edges, right-click a geometry and select **Virtual Operations>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.

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 such that it introduces closed or periodic composite edges.

Form Composite Faces

To form virtual composite faces, right-click a geometry and select **Virtual Operations>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.

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.

Collapse Edges

To virtually collapse edges into vertices, right-click a geometry and select **Virtual Operations>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.

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 that have the following properties:

- Each adjacent vertex is adjacent to more than one other edge.
- No other edge shares the same adjacent vertices (unless this is in the selection).

Merge Vertices

To virtually merge one vertex adjacent to an edge with the other adjacent vertex, right-click a geometry and select **Virtual Operations>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.

VERTEX TO REMOVE

Select the vertex that you want to remove in the Graphics window. It then appears in the **Vertex to remove** list.

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.

Backward Compatibility

If you open a model created in the 3.5a version of COMSOL Multiphysics one Composite Object feature node appears for each nonprimitive geometry object in the model.

Composite Object

The **Composite Object** node contains the follow section:

COMPOSITE OBJECT

If you save the model as a .java file the feature 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 the feature puts 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.

3

Meshing Features

This chapter describes the features available for meshing the geometry.

In this chapter:

Adding Meshing Sequences

Adding a New Meshing Sequence

When you add a new model to the Model Tree, a meshing sequence is added by default in the node **Mesh 1**. You can add more meshing sequences to the model by right-clicking the model node and selecting **Mesh**. When a model has more than one meshing sequence, they are collected under a **Meshes** node. To add new meshing sequences to such a model, you can alternatively right-click **Meshes**.

You create a mesh by building a meshing sequence.

Building Meshing Sequences

In this section:

- Physics-Controlled Meshing
- Adding and Building Meshing Features
- Editing and Building Meshing Features
- Errors and Warnings
- Deleting, Disabling, and Enabling Meshing Features
- Exporting to a File

See Also

- Meshing in the *COMSOL Multiphysics User's Guide*
- Meshing API in the *COMSOL Java API Reference Guide*

Physics-Controlled Meshing

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. If you want to modify the overall element size of the physics-induced mesh you select a new element size setting in the **Element size** list and rebuild the mesh. If you change the physics settings in your model and rebuild the meshing sequence, COMSOL Multiphysics creates a new mesh adapted to the new physics settings.

Note

A physics-induced mesh is not adapted by numerical error estimates—that type of adaptive meshing is provided by mesh adaption in the solver sequence.

To edit the physics-induced sequence, select **User-controlled mesh** in the **Sequence type** list or right-click the mesh node in the Model Tree and select **Edit Physics-Induced Sequence** ( ).

Note

By doing this the sequence is no longer updated according to changes that applied to the physics settings in the model.

If the Mesh node is right-clicked and **Reset to the Physics-Induced Sequence** ( ) is selected, 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 feature to the sequence the type of the sequence automatically switches to **User-controlled mesh**.

Adding and Building Meshing Features

You can add and build meshing features in two ways:

- By right-clicking a mesh node in the Model Tree and then selecting one of the available features. Enter the feature's properties in the **Settings** window that appears. In numerical fields, you can enter expressions that contain parameters defined in **Parameters** under the **Global Definitions** node in the Model Tree. Click the **Build Selected** button ( ) in the **Settings** window to see the mesh that results. All features can be added and built in this way.
- By using the buttons in the Mesh toolbar. Most features can be added and built in this way.

Tip

For a description of how to use the buttons in the Mesh toolbar, see the section The Mesh Toolbar; the remainder of this section discusses the first approach.

ADDING A FEATURE AT AN ARBITRARY POSITION

When you add a feature, COMSOL Multiphysics inserts it as a node in the meshing sequence after *the current feature node*. To indicate the current feature node, it appears with a quadratic frame around its icon. To add a feature after an existing feature node, first select the existing feature node and click **Build Selected** (🏢), or right-click the existing feature node and select **Build Selected**. The selected node then becomes current. Then add the feature.

Editing and Building Meshing Features

To edit a feature, select it in the tree, and make changes in the **Settings** window that appears. Features nodes that you have edited display with an asterisk (*) at the upper-right corner of their icons in the **Model Builder** window. Following features are marked with a yellow triangle at the lower right corner of the feature's icon, to indicate that they need rebuild. To see the result of your edits in the graphics, you need to build the feature. 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 features (if needed) from the first up to the selected feature.
- Click the **Build All** button (🏢) in the **Settings** window, or right-click the **Geometry** node in the tree and select **Build All**. This builds all features in the sequence (if needed).

The Graphics window shows the resulting mesh from the features that have been built. The result of subsequent features is not visible. The last built feature node becomes the *current feature node* and appears with a quadratic frame around the node's icon. The frame is green if the feature node and all preceding feature nodes are built; that is, the mesh in the Graphics window is up to date. The frame is yellow if the feature node or some preceding feature node has been edited since the feature node was built and needs to be rebuilt.

Errors and Warnings

If a problem occurs when you build a feature 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** feature node where problems are encountered, the problems related to meshing of faces and domains are avoided by leaving the corresponding faces and domains unmeshed. The operation corresponding to the feature node continues meshing the remaining entities and stores information on the encountered problems in subnodes of the feature node. A feature node that encountered this type of problems during the build gets Warning status. The feature node's icon is decorated with a yellow triangle in the lower right corner. If you build several features in a sequence the build does not stop by a feature that gets Warning status.

STOPPING THE BUILD

When you build other feature nodes than the **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 feature gets 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 feature node. If the feature is part of a sequence build, the build stops and the preceding feature node becomes the current feature node.

Deleting, Disabling, and Enabling Meshing Features

- To delete a feature node, right-click it and select **Delete** (✖).
- To disable a feature node, right-click it and select **Disable** (⊘). The disabled node does not affect the mesh. This is indicated with a gray icon and **[ ]** (brackets) around the node name in the model tree.
- To enable the feature node, right-click it and select **Enable** (◉).

To delete the entire meshing sequence, right-click the **Mesh** node (▦) and select **Delete Sequence** (✖).

Exporting to a File

To export a mesh, right-click a mesh node and select **Export**. Then browse to a file name and click **Save**.

Meshing Features

A mesh is the result of building a meshing sequence. A meshing sequence consists of Operation Features and Attribute Features. The attribute features stores properties that are used by the operation features when creating the mesh.

Building an operation feature creates or modifies the mesh on the part of the geometry defined by the selection of the operation feature. Some of the operation features use properties defined by attribute features; for example, the Free Tetrahedral feature reads properties from the Distribution and Size attribute features. For some operation features it is possible to add local attribute features. You can add a local attribute feature node from the right-click menu of the node corresponding to an operation feature. The added local attribute feature appears in a subnode of the operation feature node. Properties defined in local attribute features of an operation feature override the corresponding properties defined in global attribute features (on the same selection).

An attribute feature contains properties defined on a selection. You can add an attribute feature as a node in the meshing sequence (this is referred to as a *global attribute feature node*) or add it as a node under an operation feature node (this is referred to as a *local attribute feature node*). Global attribute feature nodes are used by subsequent operation feature nodes when building the meshing sequence. Local attribute feature nodes are only used by the owning operation feature node.

OPERATION FEATURES

The following table provides an overview of the Operation Features:

TABLE 3-1: OPERATION FEATURES

NAME	BUTTON OR ICON	DESCRIPTION
Boundary Layers	⬭	Create boundary layer meshes
Convert	□☒	Convert elements in a mesh
Copy Edge	⬚⬚ or ⬚⬚	Copy a mesh between edges
Copy Face	▨	Copy a mesh between faces
Copy Domain	☒☒ or ▨▨	Copy a mesh between domains
Edge	⬚	Create an edge mesh

TABLE 3-1: OPERATION FEATURES

NAME	BUTTON OR ICON	DESCRIPTION
Free Quad		Create an unstructured quadrilateral mesh
Free Tetrahedral		Create an unstructured tetrahedral mesh
Free Triangular		Create an unstructured triangular mesh
Import		Import a mesh
Mapped		Create a mapped mesh
Reference		Reference to other meshing sequence
Refine		Refine a mesh
Swept		Create a swept mesh

ATTRIBUTE FEATURES

The following table provides an overview of the Attribute Features:

TABLE 3-2: ATTRIBUTE FEATURES

NAME	BUTTON OR ICON	DESCRIPTION
Boundary Layer Properties		Boundary layer properties
Distribution		Mesh distribution properties
Edge Groups		Edge groups used by Mapped feature
Edge Map		Edge map used by Copy Face and Copy Domain features
One-Point Map		One-point map used by Copy Face and Copy Domain features
Scale		Scale the element size of a mesh
Size		Mesh size properties
Two-Point Map		Two-point map used by Copy Face and Copy Domain features

BACKWARD COMPATIBILITY FEATURE

If you open a model created in the 3.5a version of COMSOL Multiphysics a **Mesh Object** feature node representing the mesh appears in the meshing sequence to handle Backward Compatibility.

Operation Features

The following sections describe the operation feature nodes and their user interfaces as briefly described in the Meshing Features section.

- Boundary Layers
- Convert
- Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D)
- Edge
- Free Quad
- Free Tetrahedral
- Free Triangular
- Import
- Mapped
- Reference
- Refine
- Swept

Boundary Layers

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 to resolve the thin boundary layers along the no-slip boundaries.

2D	In 2D, a layered quadrilateral mesh is used along the specified no-slip boundaries.

3D	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 arbitrary type can also be inserted into the layers if needed.

Boundary layers are not allowed on isolated boundaries, that is, boundaries with the same domain on each side of the boundary.

To create a boundary layer mesh, 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 the level of the geometry 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.

ADVANCED SETTINGS

2D	To introduce a boundary layer split at vertices in 2D, select the **Boundary layer split at sharp corners** check box.

3D	To introduce a boundary layer split at edges in 3D, select the **Boundary layer split at sharp edges** check box.

Select the boundary layer split where the angle between the adjacent boundary layer boundaries are greater than or equal to the value specified in the **Minimum split angle** field. Control the maximum angle of the elements in the split region by the **Maximum angle per split** parameter.

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.

See Also

- Boundary Layer Properties

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 also possible to convert the entire mesh to tetrahedral mesh. This is useful because there are a few computations and features, 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 feature 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, right-click a 2D or 3D mesh node and select **Convert** ( ) in the **More Operations** submenu. Then use the following sections to specify the parts of the mesh to convert and the method that the conversion feature 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.

Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D)

Add a **Copy Domain** node in 2D/3D (▨▨/▨▨), a **Copy Face** node in 3D (▨) or a **Copy Edge** node in 2D/3D (□□/□□) to create identical meshes on domains, boundaries or edges.

It is possible to copy a mesh from one or several source domains, boundaries or edges onto one or several destination domains, boundaries or edges. One-to-one and many-to-one **Single Destination** and many-to-many **Array Copy** (array of one-to-one operations) are allowed copy types. 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. However, a constant scaling factor is allowed. As an exception, **Copy Edge** in 3D can also be used for destination edges of different shapes.

To copy a mesh between domains, boundaries or edges, right-click a mesh node and select **Copy Domain** (in 2D or 3D), **Copy Face** (3D only), or **Copy Edge** (in 2D or 3D) in the **More Operations** submenu. Then enter the properties for the copy meshing operation using the following sections:

SOURCE DOMAINS/BOUNDARIES/EDGES

Activate the selection list by selecting its **Activate Selection** button (⏻), and select the domains/boundaries/edges to copy the mesh from in the **Graphics** window.

> Note
>
> The source domains/boundaries/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.

DESTINATION DOMAINS/BOUNDARIES/EDGES

Activate the selection list by selecting its **Activate Selection** button (⏻), and select the domains/boundaries/edges to copy the mesh to in the **Graphics** window.

TYPE OF COPY

Select **Automatic** to let the software determine the proper copy method (this is the default), 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.

> Note
>
> **Array Copy** can be used only if bijective transformation of source to destination can be found (a transformation that sets 1-to-1 mapping between source and destination).

SWITCH SELECTIONS

Press Switch Source and Destination button to switch source and destination selections. Edge or vertex map (available only for **Copy Domain** and **Copy Face** features) is also switched, if provided.

ORIENTATION (ONLY AVAILABLE FOR COPY EDGE)

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 direction arrows** in the **View** node under the **Definitions** node to view the arrow direction

In the case of several source edges the orientation refers to the direction between the destination edge and the source edge with the lowest edge index.

If you want to control the orientation of the source mesh on the destination when using the **Copy Domain** or **Copy Face** features you need to add an Edge Map, a One-Point Map, or a Two-Point Map as a local attribute. To add a local attribute feature, select one of the features in the right-click menu of the **Copy Domain** or the **Copy Face** node.

Copying a face mesh in 3D is only possible if the destination face is not adjacent to any meshed domain. The copy feature overwrites any existing mesh on the destination face.

Copying a mesh to a destination domain in 2D or face in 3D that is adjacent to a meshed domain (2D) or face (3D) is possible if the edges between these domains or faces have the same number of elements as the corresponding source edges. The mesh on the destination edges is kept and the copied domain or face 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 or faces are allowed to be partitioned differently, but only in such a way that several edges of the source domain or face 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).

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 features.

To create an edge mesh, right-click a mesh node and select **Edge** in the **More Operations** submenu. 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.

Model

If you have the Batteries & Fuel Cells Module, see Mass Transport Analysis of a High Temperature PEM Fuel Cell (3D): Model Library path **Batteries_and_Fuel_Cells_Module>PEMFC>ht_pem**.

If you have the Chemical Reaction Engineering Module, see Electrochemical Treatment of Tumors: Model Library path **Chemical_Reaction_Engineering_Module>Electrochemical_Engineering>tumor**.

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 features.

To create an unstructured quadrilateral mesh in 2D, right-click a mesh node and select **Free Quad**. To create an unstructured quadrilateral mesh in 3D, right-click a mesh node and select **Free Quad** in the **More Operations** submenu. 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 allow you 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.

Free Tetrahedral

Add a **Free Tetrahedral** node ( ) to create an unstructured tetrahedral mesh. You can control the number, size, and distribution of elements by using Size and Distribution subnodes.

To create an unstructured tetrahedral mesh, right-click a 3D mesh node and select **Free Tetrahedral**. Then enter 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 allow you 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.

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 features.

To create an unstructured triangular mesh in 2D, right-click a mesh node and select **Free Triangular**. To create an unstructured triangular mesh in 3D, right-click a mesh node and select **Free Triangular** in the **More Operations** submenu. 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 allow you 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.

ADVANCED SETTINGS

Here you can specify the triangulation method that is used by the feature 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.

Import

It is only possible to import a mesh if the geometry sequence is empty. 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** and **NASTRAN file** in 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

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.

Note

For information on the NASTRAN entries that COMSOL supports see `Import` in the *COMSOL Java API Reference Guide* (*Meshing API*).

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 ignore extended node points (that is, node points not in element vertices), select the **Ignore extended node points** check box (deselected by default).

Face Partitioning Properties

If the partitioning of the boundary elements in the mesh to import into faces is not complete you can use the **Face partitioning** list to control the partitioning. Select **Automatic** to let the software partition the boundary elements into faces automatically. (this is the default). 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 **Face partitioning** to **Manual**. The value in the **Maximum angle within face** field then limits the angle between any two boundary elements in the same face. The angle between neighboring boundary elements in the same face is kept less than the value in the **Maximum face neighbor angle** field.

If you select the **Detect planar faces** check box, the mesh import detects (approximately) planar faces. A planar face has an area larger than the total area of all faces times 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 mesh import removes small faces with an area that is less the mean area of all faces times the value in the **Removal of small faces** field.

Advanced 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 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 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 degrees) in the **Maximum curvature deviation in face** field. A face with constant curvature gets an area larger than the total area of all faces times the value in the **Detect constant curvature** field.

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.

For the mapping technique to work properly, the geometry must be fairly 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 vertices or isolated boundary segments.
- The shape of each domain must not differ too much from a rectangle.

For a geometry model that initially does not meet these criteria, it is often possible to modify it such that a mapped mesh can be successfully generated.

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 features used by the a Mapped feature 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 you need to add an Edge Group feature as a subnode; right-click the **Mapped** node and select Edge Groups.

To create a mapped quadrilateral mesh in 2D, right-click a mesh node and select **Mapped**. To create a mapped quadrilateral in 3D, right-click a mesh node and select **Mapped** in the **More Operations** submenu. 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.

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.

Reference

Use a **Reference** node ( ) to refer to another meshing sequence. Building a **Reference** feature runs the operation features of the referenced sequence. If you have a Scale feature node preceding a **Reference** feature node, or as a subnode to a **Reference** feature 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 feature has a **Scale** subnode, the attribute features in the expanded sequence are scaled accordingly. In some cases, such scaling of attributes cannot be done explicitly, and additional scale features are created instead.

To expand a reference, right-click a reference feature and select **Expand** ( ).

Refine

Use this feature to refine a mesh by splitting elements.

To refine a mesh, right-click a 2D or 3D **Mesh** node and select **Refine** ( ) in the **More Operations** submenu. Then use the following sections to specify the parts of the mesh to refine and the method that the refinement feature uses to refine the elements:

DOMAIN SELECTION

Define the domains where you want to refine the mesh. Choose the level of the geometry 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 Multiphysics 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).

Note

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 feature.

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).

Swept

The **Swept** feature ( ) 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 may consist of several connected faces.

Each face about a domain in a swept feature is classified as either a source face, a destination face, or a linking face. You can specify the source faces and destination faces manually, but in most cases this is not necessary because the swept meshes can identify source and destination faces from the geometry.

If the source faces are not meshed prior to the sweeping operation the swept feature 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, and 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.

You can control the number, size, and distribution of elements by using **Size** and **Distribution** subnodes. The **Swept** feature only reads properties from Size features defined on the entire geometry or on the domain level and Distribution features defined on the domain level.

To create a swept mesh, right-click a 3D mesh node and select **Swept**. Then enter the properties for the swept meshing operation using the following sections:

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

To specify the source faces directly, activate the **Source Faces** list by selecting its **Activate Selection** button (⏻), 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 by selecting its **Activate Selection** button (⏻), 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, that are meshed automatically by the Swept feature, 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 face. **Morph source to destination** means that the destination mesh is created from the source mesh by a morphing technique, and **Use rigid transformation** means that the destination mesh is created by a rigid transformation of the source mesh. The default, **Determine suitable method**, means that the algorithm automatically tries to determine a suitable method for creating the destination mesh.

ADVANCED SETTINGS

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.

Model

See Deformation of a Feeder Clamp: Model Library path **COMSOL_Multiphysics>Structural_Mechanics>feeder_clamp**.

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**.

If you have the 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**.

If you have the Chemical Reaction Engineering Module, see Steam Reformer: Model Library path **Chemical_Reaction_Engineering_Module>Heterogeneous_Catalysis>steam_reformer**.

Attribute Features

The following sections describe the attribute features and their user interfaces as briefly described in the Meshing Features section.

- Boundary Layer Properties
- Distribution
- Edge Groups
- Edge Map
- One-Point Map
- Scale
- Size
- Two-Point Map

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.

To add a Boundary Layer Properties feature as a subnode to a Boundary Layers node select **Boundary Layer Properties** in the right-click menu of the **Boundary Layers** node. 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

Use the **Number of boundary layers** field to specify the total number of boundary layers. In the **Boundary layer stretching factor** field you 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.

Use the **Thickness of first layer** list to specify the thickness of the first element layer — the layer adjacent to the corresponding boundary. If you select **Automatic** (the default setting), the thickness of the first layer is 1/20 of the local domain element height. Use the **Thickness adjustment factor** field to specify a scaling factor that multiplies this default size. Alternatively, select **Manual** from the **Thickness of first layer** list to specify the thickness of the first layer explicitly using the **Thickness** field.

Note

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 is respected. In some cases the boundary layer meshing algorithm can choose the create fewer layers than specified. If this happens a warning is printed to the Log page of the **Progress** window.

Distribution

Use the **Distribution** feature (▦) to specify the distribution of mesh elements along an edge, for example. It is possible to add **Distribution** feature nodes both as global feature nodes and as local feature 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.

See Also

- Meshing Features

To add a Distribution feature as a global node, right-click a mesh node and select **Distribution**. To add a Distribution feature as a subnode to an operation feature node, right-click an operation feature node and select **Distribution**. Then enter the properties using the following sections:

GEOMETRIC SCOPE (3D) / BOUNDARIES (2D) / DOMAIN SELECTION (1D)

In this section you 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.

Edge Groups

Use an **Edge Groups** node (▦) to specify the four groups of edges around a boundary (3D) /domain (2D) that is used to determine the Mapped mesh of the boundary/ domain.

To add an Edge Groups feature as a subnode to a Mapped node select **Edge Groups** in the right-click menu of the Mapped node. 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 by selecting its **Activate Selection** button (⏻), and select the edges for the first edge group in the **Graphics** window.

SECOND EDGE GROUP

Activate the **Second Edge Group** list by selecting its **Activate Selection** button (⏻), and select the edges for the second edge group in the **Graphics** window.

THIRD EDGE GROUP

Activate the **Third Edge Group** list by selecting its **Activate Selection** button (⏻), and select the edges for the third edge group in the **Graphics** window.

FOURTH EDGE GROUP

Activate the **Fourth Edge Group** list by selecting its **Activate Selection** button (⏻), 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 feature is mapped onto destination edge of the Edge Map feature with the specified orientation.

See Also	- Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D)

To add an Edge Map subnode to a **Copy Face** or a **Copy Domain** node right-click the node and select **Edge Map**. Then enter the properties using the following sections:

EDGES

Activate the **Source edge** list by selecting its **Activate Selection** button (⏻), and select the edge that you want to define as source edge in the **Graphics** window.

Activate the **Destination edge** list by selecting its **Activate Selection** button (⏻), 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, and select **Opposite orientation** to let the mesh of the source edge be copied to the destination edge in the opposite direction.

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 feature.

See Also	- Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D)

To add a One-Point Map feature as a subnode to a **Copy Face** or a **Copy Domain** node right-click the node and select **One-Point Map** from the context menu. Then enter the properties using the following sections:

POINT SELECTION

Activate the **Point on source** list by selecting its **Activate Selection** button (⏻), and select the point that you want to define as source point in the **Graphics** window.

Activate the **Point on destination** list by selecting its **Activate Selection** button (⏻), and select the point that you want to define as destination point in the **Graphics** window.

Scale

Use a **Scale** node ( ) to scale the properties of the Size, Distribution, and Boundary Layer Properties features. It is possible to add Scale feature nodes both as global feature nodes and as local feature nodes to Reference features. A Scale feature that exists as global feature node affects the size of the mesh elements generated by the subsequent operation features. A Scale feature that exists as a subnode to a Reference node affects the size of the mesh elements generated by the Reference feature only.

If two or more Scale features exist on the same selection, the resulting scale factor on that selection is the product of the given scale factors.

To add a Scale feature as a global node, right-click a **Mesh** node and select **Scale**. To add a Scale feature as a subnode to a **Reference** node, right-click a **Reference** node and select **Scale**. Then enter the properties using the following sections:

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.

> Note
>
> 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 then 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 feature nodes both as global feature nodes and as local feature nodes (see Meshing Features). 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 non empty 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 it cannot be removed. To override the mesh size settings defined by this Size feature you add another Size feature node to the sequence.

To add a Size feature as a global node, right-click a mesh node and select **Size**. To add a Size feature as a subnode to an operation feature node, right-click an operation feature node and select **Size**. Then enter the properties using the following sections:

GEOMETRIC SCOPE

> Note
>
> 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 you can select the physics for which the element size is calibrated.

If you select **Predefined** you can choose a predefined elements size that automatically determines the parameters in the **Element Size Parameters** section. A predefined element size can generate an extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, or extremely coarse mesh. Select **Custom** if you want to change the value for any parameters in the **Element Size Parameters** section. For a Size node that is not the default Size node; if you select **Custom**, a check box is automatically added by the feature for each field in the **Element Size Parameters** section. You need to select the check box for an field to activate the corresponding parameter.

ELEMENT SIZE PARAMETERS

You can freely specify all element size parameters. For a Size node that is not the default Size node, you need to select **Custom** in the **Element Size** section to be able to edit the parameters. The check box in front of each parameter (only available for Size nodes that are not the default Size node) determines if the parameter is active in the feature. The following parameters control the mesh element size:

- The value in the **Maximum element size** field specifies the maximum allowed element size.
- The value in the **Minimum element size** field specifies 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 parameter is not available in 1D.
- The **Maximum element growth rate** determines 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.
- The value in the **Resolution of curvature** field determines the size of boundary elements compared to the curvature of the geometric boundary. The curvature radius multiplied by the resolution of curvature, which must be a positive scalar, gives the maximum allowed element size along the boundary. A lower value gives a finer mesh along curved boundaries. This parameter is not available in 1D.
- In the **Resolution of narrow regions** field you control the number of layers of elements that are created in narrow regions (approximately). The value must be a non-negative scalar. If the value of this parameter is less than one, the mesh generator might create elements that are anisotropic in size in narrow regions.

Note

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.

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 feature (for more information see Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D)).

To add a Two-Point Map feature 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

Activate the **First point on source** list by selecting its **Activate Selection** button (⏻), and select the point that you want to define as first source point in the **Graphics** window.

Activate the **Second point on source** list by selecting its **Activate Selection** button (⏻), 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 by selecting its **Activate Selection** button (⏻), and select the point that you want to define as first destination point in the **Graphics** window.

Activate the **Second point on destination** list by selecting its **Activate Selection** button (⏻), and select the point that you want to define as second destination point in the **Graphics** window.

Backward Compatibility

If you open a model created in the 3.5a version of COMSOL Multiphysics a Mesh Object feature node representing the mesh appears in the meshing sequence.

Mesh Object

The **Settings** window for the **Mesh Object** feature node () contains the follow section:

MESH OBJECT

If you save the model as a .java file the feature 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 the feature 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.

The Mesh Toolbar

This section describes the meshing tools accessible from the Mesh toolbar which is a part of the Main toolbar.

Instead of creating and modifying a mesh by adding and building features from the right-click menu of a mesh node you can use the tools available in the Mesh toolbar. Using the buttons in the toolbar it is possible to add and build meshing features in a one-click fashion.

The Mesh toolbar is available when you have selected a meshing sequence node in the **Model Builder**. This means that the Mesh toolbar disappears if you, for example, select a meshing feature node in the **Model Builder**. You can use the **Toolbar** menu button ( ) in the Main toolbar to make the Mesh toolbar for a specific mesh node available; 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.

The Mesh toolbar for a 3D meshing sequence contains the following buttons:

GEOMETRIC SCOPE

Use the **Select Geometric Scope** button ( ), with the menu items **Select Domains**, **Select Boundaries**, **Select Edges**, and **Select Points**, to switch between domain, boundary, edge, and point selection mode. The selection mode determines which tools in the Mesh toolbar that are enabled.

DOMAIN MESHING TOOLS

- Use the **Free Tetrahedral** button ( ) in domain selection mode to create an unstructured tetrahedral mesh. To create an unstructured tetrahedral mesh for a domain selection, select the domains in the **Graphics** window, click the menu arrow associated with the menu button, and select 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 feature node, using the selected domains, with a Size feature, using the selected predefined element size, added as a subnode. Alternatively, you can click the button associated with the menu button. Then COMSOL Multiphysics 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.

- Use the **Swept** button ( ) to 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.

BOUNDARY MESHING TOOLS

- Use the **Free Triangular** button ( ) in boundary selection mode to create an unstructured triangular mesh. This menu button works in the same way as the **Free Tetrahedral** button.
- Use the **Free Quad** button ( ) in boundary selection mode to create an unstructured quadrilateral mesh. This menu button works in the same way as the **Free Tetrahedral** button.
- Use the **Mapped** button ( ) in boundary selection mode to create a structured quadrilateral mesh. This menu button works in the same way as the Free Tetrahedral menu button.
- Use the **Copy Face** button ( ) in boundary selection mode to copy a mesh between boundaries. If you select both the boundaries to copy the mesh from and the boundaries to copy the mesh to in the **Graphics** window and click this button the software copies the mesh by adding and building a Copy Face feature node with the source boundaries set to the selected boundaries with a mesh and the destination boundaries set to the selected boundaries without a mesh.

EDGE MESHING TOOLS

- Use the **Edge** button ( ) in edge selection mode to create an edge mesh. This menu button works in the same way as the **Free Tetrahedral** button.

OTHER TOOLS

- Use the **Boundary Layers** button ( ) in boundary selection mode to create a boundary layer mesh. If you select the boundaries where you want to insert boundary layer elements in the **Graphics** window and click this button the software inserts boundary layer elements for the selected boundaries by adding and building a Boundary Layers feature node, with the meshed domains adjacent to the selected boundaries as domain selection, with an added Boundary Layer Properties feature, using the selected boundaries, as a subnode.

- Use the **Convert** button ( ) in domain or boundary selection mode to convert a mesh. If you select the domains or boundaries in the **Graphics** window for which you want to convert the elements and click this button the software converts the mesh on the selected entities by adding and building a Convert feature node using the selection.

GEOMETRIC MEASUREMENTS

To measure the volume, area, or length of a selected domain, face, or edge, respectively, click the **Measure** button ( ). The result appears 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 available in the 3D toolbar.

You do not need to confirm the entity selections in the **Graphics** window when working with the Mesh toolbar; it is sufficient to highlight them, a state that is indicated by the color red.

The Statistics Page

Use the Statistics page to view statistics of a mesh. To open the Statistics page for a mesh, right-click the mesh node and select **Statistics** (📊). This page 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** to select 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** to select 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** to select 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** to select 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.

ELEMENT QUALITY HISTOGRAM

This section displays a histogram plot of the element quality for the specified element type and selection. The quality of an element is a value between 0 and 1, where 0.0 represents a degenerated element and 1.0 represents a completely symmetric element.

4

Definitions

This chapter describes the available functionality on the Global Definitions and Definitions branches. It also includes a description of the Materials nodes. The chapter has the following sections:

Global and Local Definitions

Depending on the geometric scope, there are two types of definitions that can be defined in the Model Builder—**Global Definitions** and local **Definitions**.

In this section:

- Global Definitions
- Local Definitions
- About Building Expressions

Global Definitions

Right-click **Global Definitions** ( ) to add the following:

- **Parameters** (P_i): global, scalar values that are used to parameterize any part of the model.
- **Variables** (a=): expression variables that are used anywhere to simplify the specifications of some properties.
- **Functions** are predefined function templates for common function types such as step functions, ramps, and random functions.

See Also

- Parameters
- Variables
- Functions

Local Definitions

After adding any Model node to the Model Builder, right-click **Definitions** ( ) to add definitions with a local scope that are applied to that specific model. Add:

- **Variables** (a=): expression variables used to simplify the specifications of some properties.
- **Selection** ( ): create a user-defined set of geometric entities for reuse throughout the model.
- **View** ( ): create a user-defined view to visualize the model.

- **Contact Pair** ( ): specify two selections of boundaries that cannot penetrate each other under deformation. Requires the use of pairs and is only available if a license includes the Structural Mechanics Module or the MEMS Module.
- **Identity Pair** ( ): specify two boundary selections that overlap but belong to different parts of a pair. Then assign a boundary condition to connect the physics in the two parts in an interface. Identity pairs are only available and necessary when the model geometry is an assembly.
- **Functions** are predefined function templates for common function types such as step functions, ramps, and random functions.
- **Model Couplings**. The coupling operators are useful for creating integrated quantities and identity mappings, adding nonlocal couplings, and many other applications.
- **Coordinate Systems**. Create different coordinate systems for use in some of the physics interfaces.
- **Probes**. To monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency, parametric).
- **Perfectly Matched Layer**
- **Infinite Element Domain**

See Also

- Variables
- Selections
- Views
- Pairs
- Functions
- Model Couplings
- Coordinate Systems
- Probes

About Building Expressions

When defining any expression, whether it is global or local, there are many available variables, mathematical functions, physical constants, and operators that can be used to define the expression.

Many common mathematical and logical operators and functions can also be used along with constants to specify parameters and variables as well as equation coefficients and material properties.

- Variables
- Physical constants are predefined constants for universal constants such as the universal gas constant and the speed of light in vacuum.
- Operators
- Mathematical functions

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

Under the **Global Definitions** node (☰), enter values in the **Parameters** table to define parameters used throughout the model making it possible to parameterize, for example, a geometric dimension. Parameters are scalar numbers that are the same for all geometries and domains, can depend on other parameters, and can contain any built-in mathematical function.

Model	• Process Control Using a PID Controller: Model Library path **COMSOL_Multiphysics>Multidisciplinary_Models>pid_control**

PARAMETERS

In the **Parameters** table or the fields under the table, enter parameter **Names**, **Expression**, and **Descriptions** manually or import from a text file. Organize the table with the **Move Up** (↑), **Move Down** (↓), or **Delete** (✖) buttons or right-click the cell and select **Move Up**, **Move Down**, or **Delete**. The **Value** column shows the value of the parameter in the base unit system.

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 to **Save** the text file. The information is saved in space-separated columns in the same order as displayed on screen.

Import or load data in text 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.

See Also	• Global Parameters in the *COMSOL Multiphysics User's Guide*

Variables

The **Variables** feature enables you to create symbolic *variables* that are defined using an expression. This section describes the syntax and names for variables that you define or that are predefined in a COMSOL Multiphysics model. Variables can be applied both at the global and local levels.

You can define *global variables* or *local variables* in the same way as parameters and functions. Global variables are applied to all the models in a file, whereas local variables are valid in the entire model (geometry) or for specific domains, boundaries, edges, or points. This provides the flexibility to create variables with different meanings in different parts of the model.

To add variables to the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Variables**.

GEOMETRIC SCOPE (LOCAL DEFINITIONS ONLY)

Select the scope from the **Geometric entity level** list—**Entire Geometry**, **Domain**, **Boundary**, **Edge** (3D only), and **Point**.

VARIABLES

In the **Variables** table or the fields under the table, enter variable **Names**, **Expression**, and **Descriptions** manually or import from a text file. Organize the table with the **Move Up** (⇧), **Move Down** (⇩), or **Delete** (✖) buttons or right-click the cell and select **Move Up**, **Move Down**, or **Delete**.

Save the variables to a text file to reuse in other models with the **Save to File** button (💾). The information is saved in space-separated columns in the same order as displayed on screen.

Import or load data in text files from a spreadsheet program, for example, with the **Load from File** button (📂). Data must be separated by spaces or tabs.

See Also

In the *COMSOL Multiphysics User's Guide*:

- About Global and Local Definitions
- Operators, Functions, and Variables Reference
- Variables

Functions

In this section:

About Global and Local Functions

There are three broad categories of functions—**Analytic**, **Interpolation**, and **Piecewise**—and a number of templates for common function types, such as step and ramp functions. See Table 4-1for a list. Also create external function interfaces to include functions written in C and MATLAB. Functions can be global or local in scope, although two function types—external functions and MATLAB functions—can only be defined globally.

To add functions to the **Model Builder**, right-click **Global Definitions** or **Definitions** and select a **Function**.

OTHER FUNCTION CHARACTERISTICS

Smoothing

Many functions type have smoothing as an optional choice. Smoothing makes a function more well-behaved and more suitable for use in finite element modeling. It replaces jumps with smooth transitions that eliminates discontinuities and can represents the inertia present in real applications.

Plotting

Click the **Plot** button () in the upper-right corner of the **Settings** window to create plots of all functions of 1–3 variables. For analytic functions, first define a range for the arguments in the **Plot Parameters** section.

Click the **Create Plot** button () in the upper-right corner of the **Settings** window to create a customized plot of the function under Results, including default plot groups and plots.

Adding Functions to Materials

Define **Analytic**, **Interpolation**, and **Piecewise** functions for materials. See Materials in the *COMSOL Multiphysics User's Guide*. Then follow the instructions as described in this section.

Units for Function Inputs and Outputs

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, define the 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. In the **Function** field, type the unit for the function's output. See Using Units in the *COMSOL Multiphysics User's Guide* for general information about available units.

TABLE 4-1: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Analytic Functions	x^2 sin(x)	An analytic function is defined by a symbolic expression. It extends variables with the ability to bind parameters during function calls. In other words, the actual variable names in an expression are not required when writing the function. In addition to the arguments that are defined, analytic functions also recognize global parameters.
Elevation Functions (DEM)		The elevation function reads elevation data from files with digital elevation models (DEM files using the United States Geological Survey geospatial file format).
External Functions (Global Only)	f(x)	An external function interfaces to other external functions written in COMSOL. Then use those functions as any other functions defined in COMSOL Multiphysics. For example, for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file).
Gaussian Pulse Functions		The Gaussian pulse function is the common bell-shaped curve. It has a shape that is similar to a Gaussian (normal) distribution.
Image Functions		The image function provides a single color channel value as a 2D function of the space coordinates for any image in the BMP, JPEG, PNG, or GIF format.
Interpolation Functions		An interpolation function is defined by a table or file containing the values of the function in discrete points. The interpolation data can be both structured, (defined on a grid) or unstructured (defined on a generic point cloud).
MATLAB Functions (Global Only)	f(x)	Use a MATLAB function feature to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL Multiphysics. MATLAB functions are evaluated in MATLAB.

TABLE 4-1: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Piecewise Functions	∧	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. This function has one variable with different definitions on different intervals, which must not overlap or have any holes between them.
Ramp Functions	⟋	A ramp function is a linear increase with a user-defined slope that begins at some specified time. This function has one variable (the time t, for example).
Random Functions	∿	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. To generate a random function of the space 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 using during a parametric sweep, for example.
Rectangle Functions	⊓	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; 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. This function has one variable (the time t, for example).
Step Functions	⌐	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. This function has one variable (the time t, for example).

TABLE 4-1: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Triangle Functions	Λ	A triangle function is a linear increase and linear decline within an interval and 0 everywhere else. Use the triangle function for band-pass filtering; that is, use it to select values within an interval. This function has one variable (the time *t*, for example).
Waveform Functions	∿	A waveform function is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. This function has one variable (the time *t*, for example).

See Also

In the *COMSOL Multiphysics User's Guide:*

- Specifying Discontinuous Functions
- Operators, Functions, and Variables Reference
- Using Units
- Materials
- Global and Local Functions

Analytic Functions

An **Analytic** function is defined by a symbolic expression. It extends variables with the ability to bind parameters during function calls. In other words, analytic functions do not require the actual variable names in an expression when writing the function.

Model

If you have the AC/DC Module, see Geoelectrics: Model Library path **ACDC_Module>General_Industrial_Applications>geoelectrics**.

FUNCTION NAME

Enter a name for the function.

PARAMETERS

In the **Expression** field, enter the mathematical expression that defines the functions, such as `sin(x)*cos(y)`. Enter **Arguments** to the analytic function as comma-separated entries (`x, y` for the function above). In addition to the arguments that are defined, analytic functions also recognize global parameters. From the **Derivatives** list, select **Automatic** to compute the derivatives symbolically. Select **Manual** to enter the derivatives with respect to one or more of the arguments to the function. For undefined derivatives, 0 is used as the value of the derivative.

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. Define the interval in the **Lower limit** and **Upper limit** fields.

UNITS

Specify units for the function's inputs in the **Arguments** field and for the function's output in the **Function** field, if desired. See Units for Function Inputs and Outputs for details.

ADVANCED

Select the **May produce complex output for real arguments** check box if the symbolic expression can evaluate to a complex number for real arguments. The square root function is an example of a function that can produce complex-valued output with a real input.

PLOT PARAMETERS

Generate a preview plot of the function and set ranges for all the arguments. For each row in the **Plot Parameters** table, enter the **Argument** name, a **Lower limit**, and an **Upper limit**.

Click the **Plot** button () to generate a preview plot of the function and set ranges for all the arguments. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Elevation Functions

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.

FUNCTION NAME

Enter a name for the function in the **Function name** field.

FILE

Enter the path and name of the elevation file in the text field, or click **Browse** to select a DEM file with elevation data in the **Elevation Data** dialog box. Then click **Import** to import the elevation data into the model; otherwise COMSOL Multiphysics references the elevation data on your file system. When you have imported the elevation data, in the **File** section, under **Data imported into model**, contains information about the filename and the location for the data. In the **Export** area you can type a filename or **Browse** to find one. Then click **Export** to save the elevation data to a file and reference from there instead of including it in the model.

INTERPOLATION AND EXTRAPOLATION

For interpolation in the elevation data, select one of the following interpolation methods from the **Interpolation** list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the elevation data, select one of the following extrapolation methods from the **Extrapolation** list—**Constant** (the default), **Linear**, **Nearest function** or **Specific value**. For this case, type the value in the **Value outside range** field (default value: 0).

Click the **Plot** button ( ) to generate a preview plot of the elevation function using a surface plot with height to represent the output from the elevation function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

External Functions (Global Only)

Note

The **External** function is only available globally on the **Global Definitions** node. See also model.func() as described in the *COMSOL Java API Reference Guide* for details about the interface to the library.

An **External** function interfaces to other external functions written in COMSOL. Then use those functions as any other functions defined in COMSOL Multiphysics. For example, use it for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file).

FUNCTIONS

In the **Library** field, enter a **Library** (the complete network path) or **Browse** to locate a library to import. For each row in the **Functions** 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).

DERIVATIVES

Enter data in this section based on the following. 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** column must occur in the function 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 get the default value 0.

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). Selecting this check box can then improve performance.

The **Minimize calls** check box is selected by default. The program then tries to improve performance by minimizing the calls to the external function and the overhead associated with such function calls.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Gaussian Pulse Functions

The **Gaussian Pulse** function is the common bell-shaped curve. It has a shape that is similar to a Gaussian (normal) distribution.

FUNCTION NAME

Enter a name for the function.

PARAMETERS

Enter a **Location** value for the Gaussian pulse mean s_0. Enter a **Standard deviation** σ of the normal distribution.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Image Functions

The **Image** function ( ) makes it possible to import an image (in BMP, JPEG, PNG, or GIF format) and map the image's RGB data to a scalar (single channel) function output value. By default the function's output uses the mapping (R+G+B)/3.

FUNCTION NAME

Enter a name for the function in the **Function name** field.

FILE

Enter the path and name of the image file in the text field, or click **Browse** to select an image file in the **Image** dialog box. Then click **Import** to import the image into the model; otherwise COMSOL Multiphysics references the image on your file system. When you have imported the image, the **File** section contains information about the image's filename and size. In the **Export** area you can type a filename or **Browse** to find one. Then click **Export** to save the image to a file and reference from there instead of including it in the model.

COORDINATES

Select the In place check box to use the pixels in the image as the coordinates. Clear this check box to defined 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 the same expression as for the automatic setting: `(r+g+b)/3`.

For interpolation in the image, select one of the following interpolation methods from the **Interpolation** list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the image, select one of the following extrapolation methods from the **Extrapolation** list—**Constant** (the default), **Linear**, **Nearest function**, or **Specific value**. For this case, type the value in the **Value outside range** field (default value: 0).

CLIPPING

You can apply clipping to create a box-shaped region inside of the original image where the image will be rendered. From the **Clipping** list, select:

- **None** (the default) for no clipping.
- **Manual** to define the box-shaped region using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields.

UNITS

Specify units for the function's inputs (the *x* and *y* coordinates) in the **Arguments** field (default: m) and for the function's output in the **Function** field (default: no unit), if desired. See Units for Function Inputs and Outputs for details.

Click the **Plot** button ( ) to generate a preview plot of the image function using a surface plot with height to represent the output from the image function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Interpolation Functions

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).

Model

If you have the Acoustics Module, see:

- Muffler with Perforates: Model Library path **Acoustics_Module>Industrial_Models>perforated_muffler**
- Open Pipe: Model Library path **Acoustics_Module>Benchmark_Models>open_pipe**

Note

See Interpolation Functions in the *COMSOL Multiphysics User's Guide* for extensive details about this page.

PARAMETERS

Select a **Data source**—**Table** 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. 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. Alternatively, import or load data in text files from a spreadsheet program, for example, with the **Load from File** button (📂). Data must be separated by spaces or tabs. In case that you also want to use the inverse function $t = f^{-1}(y)$, select the **Define inverse function** check box and enter a function name for the inverse function in the **Inverse function name** field (the default name is the default function name with the suffix **_inv**). You can only define the inverse function for interpolation functions that are defined using a table.

If **File** is selected to import function data, select a **Data format**—**Spreadsheet**, **Grid**, or **Sectionwise**. If **Spreadsheet** is selected, enter a **Number of arguments**. 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; the following columns can contain one or more function, and the positions entered are the relative position for each function's data column.

For the common case where the data source contains function values that are functions of the space coordinates, select the **Use space coordinates as arguments** check box. Then the function can be called without arguments when used in the model; the space coordinates are added as function arguments automatically.

INTERPOLATION AND EXTRAPOLATION

The interpolation setting controls 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**, **Piecewise cubic**, or **Cubic spline**. For functions of more than one variable, select **Nearest neighbor** or **Linear**.

Select an **Extrapolation** method: **Constant**, **Linear** (**Piecewise cubic** or **Cubic spline** must be selected from the **Interpolation** list), **Nearest function**, or **Specific value**. Enter the value in the **Values outside range** field.

UNITS

Specify units for the function's inputs in the **Arguments** field and for the function's output in the **Function** field, if desired. See Units for Function Inputs and Outputs for details.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

MATLAB Functions (Global Only)

Important	A COMSOL LiveLink for MATLAB license is required. Also confirm that MATLAB is installed on the same computer as COMSOL Multiphysics.

Use **MATLAB** functions node f(x) from COMSOL Multiphysics to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL Multiphysics. MATLAB functions are evaluated in MATLAB.

Note	See Using MATLAB Functions in COMSOL Multiphysics in the *LiveLink for MATLAB User's Guide* for more information about using MATLAB functions.

FUNCTIONS

Under **Functions**, 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).

DERIVATIVES

Enter data in this section based on the following:

- 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** column must occur in the function 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.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Piecewise Functions

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. This function has one variable with different definitions on different intervals, which must not overlap or have any holes between them.

FUNCTION NAME

Enter a name for the function.

PARAMETERS

Enter a piecewise function name in the **Argument** field. Select an **Extrapolation** method: **Constant**, **None**, **Nearest function**, or **Specific value**.

Select a **Smoothing**: **No smoothing**, **Continuous function**, **Continuous first derivative**, or **Continuous second derivative**. If **Continuous function**, **Continuous first derivative**, or **Continuous second derivative** is selected, enter a value in the **Relative size of transition zone** field.

For each cell in the **Intervals** table, enter **Start** and **End** interval limits. The intervals must not overlap, there cannot be any holes between intervals. Enter an expression defining the **Function**.

UNITS

Specify units for the function's inputs in the **Arguments** field and for the function's output in the **Function** field, if desired. See Units for Function Inputs and Outputs for details.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Ramp Functions

A **Ramp** function is a linear increase with a user-defined slope that begins at some specified time. This function has one variable (the time t, for example).

FUNCTION NAME

Enter a name for the function.

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. Enter a **Slope** k of the ramp. To ensure that the value never exceeds a certain point, select the **Cutoff** check box and enter a value. 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 $s\geq s_0$.

SMOOTHING

To turn on the smoothing function at the start or the cutoff, select the **Smooth at start** or **Smooth at cutoff** check boxes. Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. Smoothed ramp functions have continuous first and second derivatives.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Random Functions

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.

To generate a random function of the space 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 using during a parametric sweep, for example.

FUNCTION NAME

Enter a name for the function.

PARAMETERS

Enter a **Number of arguments** to the random function. Select **Uniform** or **Normal** from the **Distribution** list. If **Uniform** is selected, enter a **Mean** and **Range**. 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**.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Rectangle Functions

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; 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. This function has one variable (the time *t*, for example).

Model

- Fluid Valve: Model Library path **COMSOL_Multiphysics>Fluid_Dynamics>fluid_valve**
- 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**.

FUNCTION NAME

Enter a name for the function.

LIMITS

Enter a **Lower limit** and **Upper limit** 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 an interval. Outside the interval it evaluates to 0.

SMOOTHING

To turn on the smoothing function select the **Size of transition zone** check box. Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Step Functions

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. This function has one variable (the time t, for example).

Model

- Fluid Valve: Model Library path **COMSOL_Multiphysics>Fluid_Dynamics>fluid_valve**

FUNCTION NAME

Enter a name for the function.

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 **End** 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 $L_{\text{end}} - L_{\text{start}}$.

SMOOTHING

To turn on the smoothing function select the **Size of transition zone** check box. Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Triangle Functions

A **Triangl**e function is a linear increase and linear decline within an interval and 0 everywhere else. Use the triangle function for band-pass filtering; that is, use it to select values within an interval. This function has one variable (the time t, for example).

FUNCTION NAME

Enter a name for the function.

LIMITS

Enter a **Lower limit** and **Upper limit** 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.

SMOOTHING

To turn on the smoothing function select the **Size of transition zone** check box. Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Waveform Functions

A **Waveform** function is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. This function has one variable (the time t, for example).

FUNCTION NAME

Enter a name for the function.

PARAMETERS

Select a waveform from the **Type** list: **Sawtooth**, **Sine**, **Square**, or **Triangle**. If **Sine** is selected, enter a **Frequency**, **Phase**, and **Amplitude** value. Smoothing is not available for this waveform. For **Sawtooth**, **Square**, or **Triangle** waveforms, the **Smoothing** check box is checked by default. Enter a value in the **Transition zone** field. If required, turn off smoothing by clearing the **Smoothing** check box. For **Sawtooth**, **Square**, or **Triangle**, enter a **Frequency**, **Phase**, and **Amplitude** value.

Click the **Plot** button () to generate a preview plot of the function. Click the **Create Plot** button () to create a plot group and a plot for the function plot.

Model Couplings

In this section:

- About Model Couplings and Coupling Operators
- General Extrusion Model Couplings
- Linear Extrusion Model Couplings
- Boundary Similarity Model Couplings
- One-Point Maps
- Two-Point Maps
- Edge Maps
- Identity Mapping Model Couplings
- General Projection Model Couplings
- Linear Projection Model Couplings
- Integration Model Couplings
- Average Model Couplings
- Maximum and Minimum Model Couplings

About Model Couplings and Coupling Operators

Model couplings establish couplings between different parts of a model or between different models. A model coupling is defined by a *coupling operator*, taking an expression as its argument. When the operator is evaluated at a point in the destination, the value is computed by evaluating the argument on the source. Use a coupling operator to compute several quantities that use the same source and destination. All types of model couplings have a source and a destination.

Note	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, while the destination can, depending on the type of coupling, be a subset of one or several models or a value that can be evaluated everywhere. The destination value is computed by evaluating the operator's argument at one or several points in the source. Both the source and the destination are purely geometrical objects, but the source is limited to one geometry whereas the destination is usually global.

Note

With extrusion and projection couplings, the final evaluation fails if the mapping fails, and therefore it is the destination that must be well defined.

To add model couplings to the **Model Builder**, right-click **Definitions** and select a **Model Coupling** from Table 4-2.

ABOUT COUPLING OPERATORS

Coupling operators are useful for modeling coupled problems and are a generalization of ordinary expressions. They are defined by first selecting the source, where the argument of the operator is evaluated, and, in some cases, a destination. An evaluation expression is not required to define coupling operators; instead these operators are used in different contexts, passing the expression to evaluate as a input argument.

There are three categories of coupling operators as listed in Table 4-2:

- *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.
- *Projection*. These coupling operators—General Projection and Linear Projection—evaluate a series of line or curve integrals on the source domain, where the line or curve positions depend on the positions of the evaluation points in the destination domain.
- *Scalar*. These operators—Integration, Average, Maximum, and Minimum—define a scalar value such as an integration, the average over a set of geometric entities, and the maximum or minimum value of an expression and have a “global destination” (that is, they are available everywhere in the model).

TABLE 4-2: COUPLING OPERATOR DESCRIPTIONS

LINK TO SECTION	ICON	DESCRIPTION
EXTRUSION MODEL COUPLINGS		
General Extrusion Model Couplings		A General Extrusion model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a general extrusion coupling to define a nonlinear mapping.
Linear Extrusion Model Couplings		A Linear Extrusion model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a linear extrusion when the correspondence between evaluation points in the source and destination is linear (and in some to nonlinear cases).
Boundary Similarity Model Couplings		A Boundary Similarity model coupling maps an expression defined on a part of a boundary to another part of a boundary with the same shape. Available in 2D and 3D, it is defined by a similarity mapping from the destination to the source. Also add Edge Map, One Point Map, or Two Point Map subnodes.
Identity Mapping Model Couplings		An Identity Mapping model coupling maps between geometric entities which 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.
PROJECTION MODEL COUPLINGS		
General Projection Model Couplings		Use a General Projection operator to define integration along curves.
Linear Projection Model Couplings		Use a Linear Projection model coupling when the argument is to be integrated along a line, and the line depends linearly on the evaluation point.

TABLE 4-2: COUPLING OPERATOR DESCRIPTIONS

LINK TO SECTION	ICON	DESCRIPTION
SCALAR MODEL COUPLINGS		
Integration Model Couplings	∫du	An Integration model coupling integrates an expression over selected geometric entities. This is evaluated by integrating the argument over the source domain or summing the argument over the node points in the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.
Average Model Couplings	∿	An Average model coupling computes the average of an expression over selected geometric entities. It can be evaluated anywhere in any model, and the value does not depend on the evaluation point. 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 dimension.
Maximum and Minimum Model Couplings	∿ ∿	These operators compute the maximum or minimum of an expression over selected geometric entities and give the maximum and minimum values of the arguments over the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

See Also

In the *COMSOL Multiphysics User’s Guide*:

- Model Couplings
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

General Extrusion Model Couplings

A **General Extrusion** model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a general extrusion coupling to define a nonlinear mapping.

Model	If you have the Acoustics Module, see Flow Duct: Model Library path **Acoustics_Module>Industrial_Models>flow_duct**. If you have the RF Module, see Radar Cross Section: Model Library path **RF_Module>Tutorial_Models>radar_cross_section**. If you have the Subsurface Flow Module, see Aquifer Characterization: Model Library path **Subsurface_Flow_Module>Fluid_Flow>aquifer_characterization**.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`genext1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

DESTINATION MAP

Specify the general extrusion destination map by entering 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.

Note	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.

SOURCE

Select a **Source frame** to use on the source domain. 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. 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.

ADVANCED

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.

Linear Extrusion Model Couplings

A **Linear Extrusion** model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a linear extrusion when the correspondence between evaluation points in the source and destination is linear (and in some to nonlinear cases).

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`linext1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

SOURCE

Select a **Source frame** to use on the source domain. 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. With the default source map expressions, the intermediate mesh can be considered identical to the source.

SOURCE VERTICES

As required, enter values for **Source vertex 1**, **2**, **3**, and **4**. The selection of **Source Vertices** and **Destination Vertices** together define the linear mapping from the destination to the source. The number of source vertices entered should be the same as the number of destination vertices. If not all source vertices selections are used, the empty selections must be last. Click the **Activate Selection** button (⏻) to define the source vertices. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** (🧹) buttons as required.

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.

DESTINATION VERTICES

As required, enter values for **Destination vertex 1**, **2**, **3**, and **4**. The number of destination vertices should be the same as the number of source vertices. If not all destination vertex selections are used, the empty selections must be last.Click the **Activate Selection** button (⏻) to define the source vertices. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** (🧹) buttons as required.

ADVANCED

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.

Boundary Similarity Model Couplings

A **Boundary Similarity** model coupling maps an expression defined on a part of a boundary to another part of a boundary with the same shape. Available in 2D and 3D, it is defined by a similarity mapping from the destination to the source.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`bndsim1`, for example).

SOURCE BOUNDARIES

Select **Manual** or **All Boundaries** from the **Selection** list to define the source selection of a boundary similarity operator. If **Manual** is selected, select geometric entities in the **Graphics** window. Select **All boundaries** to add all geometry to the **Selection** box.

DESTINATION BOUNDARY

Enter a single boundary as the destination.

ADVANCED

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.

2D

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.

3D

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 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.

One-Point Maps

Use a **One-Point Map** to control a **Boundary Similarity Model Coupling** mapping in 3D.

POINT SELECTION

Click the **Activate Selection** button (⏻) to select a **Point on source** and a **Point on destination** adjacent to the source and destination selection, respectively.

Two-Point Maps

Use a **Two-Point Map** to control a **Boundary Similarity** model coupling mapping in 3D.

SOURCE POINTS

Click the **Activate Selection** button (⏻) to select a **First point on source** and a **Second point on source**. Click in the **Graphics** window and select the points adjacent to the source selection.

DESTINATION POINTS

Click the **Activate Selection** button (⏻) to select **First point on destination** and a **Second point on destination**. Click in the **Graphics** window and select the points. The destination points are mapped to the corresponding points on the source by the similarity mapping.

Edge Maps

Use an **Edge Map** to control a **Boundary Similarity** model coupling mapping in 3D.

EDGE SELECTION

Click the **Activate Selection** button (⏻) to select a **Source edge** and a **Destination edge**. Click in the **Graphics** window and select the edges. 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**, **Same**, 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 Model Couplings

An **Identity Mapping** model coupling maps between geometric entities which 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.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`idmap1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

FRAMES

Select a **Source frame** and a **Destination frame** to use on the geometric entities. In most cases the default **Spatial** frame can be used.

ADVANCED

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.

General Projection Model Couplings

A projection coupling operator evaluates an expression defined on a source domain by integration along lines or curves depending on the evaluation point in the destination domain. Use a **General Projection** operator ( ) to define integration along curves.

Note

It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`genproj1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

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 selection. If the selection has lower dimension than the source geometry, specify only the first srcedim expressions. Use expressions containing space coordinates in the source geometry when defining the map. Remember that 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 it is permissible to use expressions containing space coordinates in the destination geometry.

ADVANCED

Enter an **Integration order** of the numerical integration method.

Linear Projection Model Couplings

Use a **Linear Projection** model 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 source domains and destination domains of the nearest lower dimension. The domains can exist in geometries of different space dimensions. For example, couple from domains in 2D to edges in 3D; or couple 3D domains to 2D domains. Define the linear projection by specifying points in both the source and destination domains.

Note

It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`linproj1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

SOURCE

Select a **Source frame** from the list to evaluate the coordinates of the source vertices in the selected frame.

SOURCE VERTICES

Select **Source Vertices** for each of the fields. Select $srcedim + 1$ source vertices where $srcedim$ is the dimension of the source selection. Depending on the dimension of the source selection, it may be that some of the last source vertex selections should be left empty.

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**.

DESTINATION VERTICES

As required, enter values for **Destination vertex 1**, **2**, and **3**. 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.

ADVANCED

Enter an **Integration order** of the numerical integration method.

Integration Model Couplings

An **Integration** model coupling integrates an expression over selected geometric entities. This is evaluated by integrating the argument over the source domain or summing the argument over the node points in the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

Model

- Acoustics of a Muffler: Model Library path: **COMSOL_Multiphysics>Acoustics>automotive_muffler**
- Fluid Valve: Model Library path **COMSOL_Multiphysics>Fluid_Dynamics>fluid_valve**

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`intop1`, for example).

SOURCE SELECTION

From the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Select **Manual** or **All Domains**, **All Boundaries**, **All Edges**, **All Points** from the **Selection** list. If **Manual** is selected, select geometric entities in the **Graphics** window. Select **All** to add the applicable geometry to the **Selection** box.

ADVANCED

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. Select a **Frame** from the list for the volume element to be used in the integration. In an axisymmetric model, select the **Compute integral in revolved geometry** check box to perform the integration in 3D (as a volume integral) for a 2D axisymmetric model or in 2D for a 1D axisymmetric model.

Average Model Couplings

An **Average** model coupling computes the average of an expression over selected geometric entities. It can be evaluated anywhere in any model, and the value does not depend on the evaluation point. 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 dimension.

Model

- Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics>Diffusion>effective_diffusivity**

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (aveop1, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

ADVANCED

Enter a value in the **Integration order** field. Select a **Frame** from the list. In an axisymmetric model, select the check box **Compute integral in revolved geometry** to compute the average in 3D for a 2D axisymmetric model or in 2D for a 1D axisymmetric model.

Maximum and Minimum Model Couplings

The **Maximum** and **Minimum** coupling operators compute the maximum or minimum of an expression over selected geometric entities and gives the maximum or minimum value of the expression in the argument over the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

Model

- Deformation of a Feeder Clamp: Model Library path **COMSOL_Multiphysics>Structural_Mechanics>feeder_clamp**

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name (`maxop1`, for example).

SOURCE SELECTION

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** to add the applicable geometry to the **Selection** box.

ADVANCED

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. From the **Point type** list, select **Node points** (the default), **Integration points**, or **Lagrange points** to compute the maximum by evaluating the argument expression at a finite set of points in the source and taking the maximum of these values. If **Integration points** is selected, enter an **Integration order**. The default is `4`. If **Lagrange points** is selected, enter a **Lagrange order**. The default is `2`.

Materials

Use the nodes under **Materials** to add predefined or user-defined materials, specify specific properties, model inputs, functions, values and expressions, or create a custom material library.

MATERIAL OVERVIEW

This section provides an overview of the materials in the model and where they are used.

The **Material** column lists the current materials in the model with their names and, within curly braces, their tags.

The **Selection** column lists the geometric entities selected for the material (the domains, boundaries, or edges where the material is defined).

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.

In this section:

- The Material Browser Window
- The Material Page
- Property Groups

See Also

In the *COMSOL Multiphysics User's Guide:*

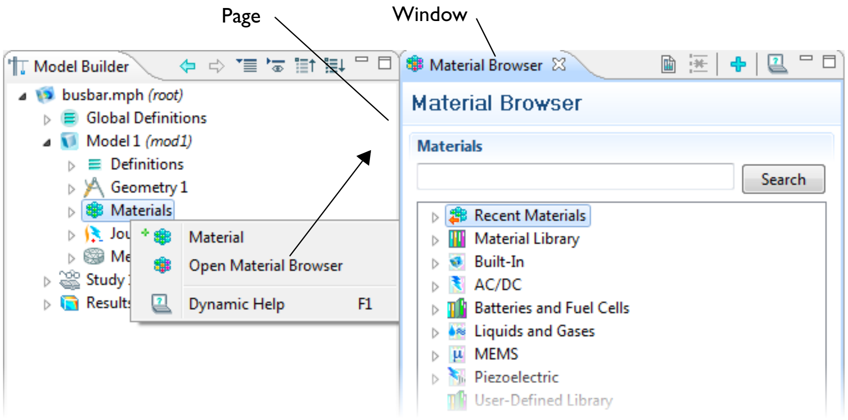
- Materials
- About Selecting Geometric Entities
- Global and Local Functions

Note

When using the Material Browser, the words *window* and *page* are interchangeable. For simplicity, the instructions refer only to the Material Browser.

The *Material Browser* window ( ) contains a number of databases with a broad collection of physical, elastic, electromagnetic, fluid, piezoelectric, and thermal properties of materials. Use the Material Browser to find predefined materials and add them to the Model Builder, or create a custom material library.

The sections in the **Material Browser** display the predefined material properties that are chosen under the **Material Selection** section.



MATERIALS

Browse all the available material databases or search in the **Materials** section. **Search** a specific material by name, UNS number, or DIN number. Also browse for a specific material by class (for example, in the Material Library) or physics module (for example, MEMS). When the material is located, right-click to **Add Material to Model**.

INFORMATION

When browsing the material databases, in particular the **Model Library**, some materials include additional information—UNS number, DIN number, and/or Composition.

PROPERTIES

While browsing the databases, predefined **Properties** are listed in a table with the default **Property**, **Expression**, and **Property groups** to which the material belongs.

FUNCTIONS

While browsing some materials, predefined **Functions** are listed. The table lists the default **Function**, **Type**, **Expression**, and **Property groups** to which the material belongs.

INPUT

While browsing some materials, predefined **Input** are listed in a table with the default **Input**, **Unit**, and **Property groups** to which the material belongs.

See Also

- Materials in the *COMSOL Multiphysics User's Guide*

When you have the Material Library:

- Material Library Quick Start
- About the Material Library Folders and Properties

The Material Page

The **Material** page ( ) summarizes the predefined or user defined material properties. It is where material properties are added or changed specific to the model and the geometric entity level is assigned where the material is used in the geometry (domain, boundary, edge, or point) and the domains, for example, that consists of this material.

After adding a material, click the material node (for example, **Air**) in the **Model Builder**. The **Material** page opens in the **Settings** window and is linked to other pages with the selected material data.

GEOMETRIC SCOPE

Assign the material to a specific **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point**—on the geometry in the **Graphics** window (the geometry in the model).

MATERIAL PROPERTIES

Add additional material properties to the model if they are not already included in the default settings.

A Note about Adding Basic Material Properties

Material properties can be added to models from two locations—the **Material** and **Property Group** pages. In both cases the properties belong to the **Basic** property group.

- When this information is accessed from the **Basic** node's **Property Group** page, it is listed under **Quantities>Output Properties**.
- When this information is accessed from the **Material** page, it is listed under **Material Properties>Basic Properties**.

MATERIAL CONTENTS

This section lists of all the material properties that are defined for the material or required by the physics interfaces in the model. The table lists the **Property**, **Name**, **Value**, **Unit**, and the **Property group** that the material property belongs to. The **Property group** corresponds to the nodes in the **Model Builder** with the same name. If required, edit the values or expression for the property's **Value**. A stop sign displays in the left column to indicate when a required property is missing a **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 interface but undefined.
- A warning sign (⚠) indicates that the material parameter is undefined. An entry is only required if the material property is 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 interface in the model.

APPEARANCE

Control or change the default appearance of a material in the Graphics window when working in the Materials or Physics interface sections.

2D

For 2D models, only a change of color is possible.

3D

For 3D models, the material appearance is with both color and texture.

The default **Family** appearance is taken from the material selected, which has a suitable color (2D and 3D models), and texture (3D models) applied. Select another option from the **Family** list if required—**Air**, **Aluminum**, **Brick**, **Concrete**, **Copper**, **Gold**, **Iron**, **Lead**, **Magnesium**, **Plastic**, **Steel**, **Titanium**, or **Water**. Select **Custom** to make further adjustments. The default custom settings are inherited from the material selected last from the **Family** list.

Note

The appearance includes texture and reflectance only when the preference settings for the visualization are optimized for quality. When optimized for performance, the appearance includes color only. Control this using the **Optimize for** list in the **Graphics** section in the **Preferences** dialog box.

See Also

In the *COMSOL Multiphysics User's Guide:*

- Materials
- Customizing the Appearance of a Material in the Graphics Window

Property Groups

The **Property Group** page is where output properties and model inputs are added, local parameters defined, and expressions for material properties entered in a specific property group such as **Basic**. The property groups are subnodes to a material node.

The **Property Group** page is in the **Settings** window and opens when the node is clicked (for example, **Basic**) under the material name in the **Model Builder**.

OUTPUT PROPERTIES AND MODEL INPUTS

View predefined material properties in the **Output properties** table. Under **Quantities** add additional material properties to the **Output properties** list or add model inputs to the **Model inputs** list. The model inputs are physical quantities such as temperature that are used as inputs in the expressions that define the output properties. For example, adding **Temperature** as a model input with the variable name `T` makes it possible to use an expression 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** subsection's **Expression** column; edit directly in the table or by clicking the **Edit** button (📝), which opens the **Edit** dialog box. For an anisotropic material property (tensor), select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** when entering the data in the **Edit** 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.

A Note about Adding Basic Material Properties

Material properties can be added to models from two pages—**Material** and **Property Group**. In both cases, the properties belong to the **Basic** property group.

- When this information is accessed from the **Basic Property Group** page, it is listed under **Quantities>Output Properties** and the **Variable** is listed in the table.
- When this information is accessed from the **Material** page, it is listed under **Material Properties>Basic Properties** and the **Name** is listed in the tables.

LOCAL PARAMETERS

Enter a user defined **Parameter** and its corresponding **Expression** and organize the table as required.

About Automatic Adding of Property Groups to a Material

Material property groups are automatically added to the material node in the **Model Builder**. Also add additional predefined property groups or create a **User defined property group**. The available properties are grouped together differently according to the physical context.

Each of the property groups has a **Property Group** page. When a **Model Builder** node is clicked (for example, **Basic**), the **Property Group** page displays specific information about that property group. The property groups are summarized in a **Material Contents** table on the **Material** page.

LIST OF PREDEFINED MATERIAL PROPERTY GROUPS

- Basic: Contains over 25 basic properties for use with all materials.
- Linearized Resistivity
- Loss Tangent

- Dielectric Losses
- Refractive Index
- Magnetic Losses
- HB Curve
- BH Curve
- Young’s Modulus and Poisson's Ratio
- Lamé Constants
- Bulk Modulus and Shear Modulus
- Pressure-Wave and Shear-Wave Speeds
- Orthotropic
- Anisotropic
- Elastoplastic Material Model
- Viscoelastic Material Model
- Neo-Hookean
- Mooney-Rivlin
- Murnaghan
- Stress-Charge Form
- Strain-Charge Form
- Ideal Gas

Note

Some of these material groups are only used by physics interfaces in the add-on modules.

See Also

In the *COMSOL Multiphysics User’s Guide:*

- Materials
- About the Output Materials Properties

Coordinate Systems

In this section:

- About Coordinate Systems
- Base Vector Systems
- Boundary Systems
- Cylindrical Systems
- Infinite Element Domain
- Mapped Systems
- Perfectly Matched Layer
- Rotated Systems
- Spherical Systems

About Coordinate Systems

COMSOL Multiphysics 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 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*.

2D

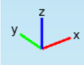
3D

For 2D and 3D models, the **Boundary System** node is automatically added under **Definitions**.

To add another coordinate system to the **Model Builder**, right-click **Definitions**, and select a **Coordinate Systems** from Table 4-3.

The spatial Cartesian coordinate system coordinates default to the following names in 2D and 3D (in 2D axisymmetric geometries, COMSOL Multiphysics uses cylindrical coordinates):

GEOMETRY	DEFAULT NAME OF SPACE 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 this 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, for example. See Table 4-3 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.

TABLE 4-3: COORDINATE SYSTEM DESCRIPTIONS

COORDINATE SYSTEM	ICON	TYPE	DESCRIPTION
Base Vector Systems		relative	1D, 2D, and 3D. Define this using a set of base vectors to form a coordinate system, which are declared as orthonormal.
Boundary Systems		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 Systems		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.
Mapped Systems		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.

TABLE 4-3: COORDINATE SYSTEM DESCRIPTIONS

COORDINATE SYSTEM	ICON	TYPE	DESCRIPTION
Rotated Systems		relative	2D and 3D. Use this to define rotation about the out-of-plane direction in 2D and Euler angles in 3D.
Spherical Systems		absolute	3D only. Use this when a field or property using spherical coordinates is to be specified.

See Also

In the *COMSOL Multiphysics User's Guide:*

- Coordinate Systems
- Space Coordinate Variables

Base Vector Systems

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.

Model

If you have the Acoustics Module, see SAW Gas Sensor (Model Library path **Acoustics_Module>Industrial_Models>saw_gas_sensor**) and Piezoacoustic Transducer (Model Library path **Acoustics_Module>Tutorial_Models>piezoacoustic_transducer**).

If you have the MEMS Module, see SAW Gas Sensor (Model Library path **MEMS_Module>Sensors>saw_gas_sensor**) and Piezoelectric Shear-Actuated Beam (Model Library path **MEMS_Module>Piezoelectric_Devices>shear_bender**).

If you have the Structural Mechanics Module, see Piezoelectric Shear-Actuated Beam (Model Library path **Structural_Mechanics_Module>Piezoelectric_Effects>shear_bender**).

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

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.

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).

Under **Simplifications**, you can set properties for the base vector systems which help simplifying the coordinate transformations. Select the **Assume orthonormal** check box if the coordinate system is orthonormal.

1D	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**.

2D *2D Axi*	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.

Boundary Systems

2D *3D*	For 2D and 3D models, a **Boundary System** node is automatically added under **Definitions**.

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.

Model

Many Model Library examples use this coordinate system. For one example, see Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics>Diffusion>effective_diffusivity**.

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

Select a **Frame type**—**Reference configuration**, or **Deformed configuration** (the default). The deformed configuration follows the material whereas the reference configuration is attached to the spatial frame.

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. 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** 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.

Cylindrical Systems

A **Cylindrical System** ( ) can be used in 2D and 3D where rotational symmetry about the axis is required.

Note

The cylindrical coordinate system is not applicable in geometries with 2D axial symmetry.

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

Select a **Frame type**—**Mesh (x,y,z)**, **Material (x,y,z)**, or **Spatial (x,y,z)** (the default), or **Geometry frame coordinate**. The frame type must match the one for the physics interface where the coordinate system is used.

In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `phi`, and `a`. In planar 2D models, `r` and `phi` are the in-plane coordinates, and `a` is the out-of-plane coordinate. Enter the **Origin of system**. The default is an origin coinciding with the one from the global system.

For 3D: Enter the **Axis direction**. The default axis direction is the z-direction in the global system. Enter the **Radial base vector direction** for azimuthal angle $\varphi = 0$. The default direction is the x-direction in the global system.

Infinite Element Domain

Note

For more information about this feature, see About Infinite Element Domains and Perfectly Matched Layers in the *COMSOL Multiphysics User's Guide*.

Use an **Infinite Element Domain** node ( ) 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 towards 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 perfectly matched layers (PMLs) (see Perfectly Matched Layer).

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

DOMAIN SELECTION

Select the domains where you want to define the infinite element domain.

SETTINGS

Based on the space dimension, select a **Frame type**—**Mesh (x, y, z)**, **Material (x, y, z)**, or **Spatial (x, y, z)**.

GEOMETRY

Select a **Type**—**Cartesian**, **Spherical**, or **Cylindrical**.

- If **Spherical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)** in the table. For axisymmetric models, only the z coordinate is required.
- If **Cylindrical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)**. For 3D models, also enter **Center axis direction** values for **x**, **y**, and **z** in each table.

SCALING

Enter an expression for the following:

- **Physical width** - the default variable is `1e3*root.mod1.dGeomChar`
- **Pole distance** - the default variable is `root.mod1.dGeomChar`

Mapped Systems

Use a **Mapped System** ( ) to create a system that defines a mapping from the frame coordinate system.

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

Select a **Frame type**—**Mesh (x,y,z)**, **Material (x,y,z)**, or **Spatial (x,y,z)** (the default), or **Geometry frame coordinate**. The frame type must match the one for the physics interface where the coordinate system is used.

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.

Under **Coordinate mapping**, the **Coordinate** column displays the **Coordinate names** with the **Expression** column displaying the associated mapped coordinate.

If required, under **Simplifications**, select the **Assume orthonormal** check box. The program then uses the assumption that the settings define an orthonormal coordinate system.

Perfectly Matched Layer

Note

For more information about this feature, see About Infinite Element Domains and Perfectly Matched Layers in the *COMSOL Multiphysics User's Guide*.

Use a **Perfectly Matched Layer** node ( ) 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 towards infinity, giving rise to infinite elements (see Infinite Element Domain). To absorb outgoing waves in a frequency-domain problem, you must instead stretch the virtual domains into the complex plane, creating perfectly matched layers (PMLs).

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

DOMAIN SELECTION

Select the domains where you want to define the perfectly matched layer.

SETTINGS

Based on the space dimension, select a **Frame type**—**Mesh (x, y, z)**, **Material (x, y, z)**, or **Spatial (x, y, z)**.

GEOMETRY

Select a **Type**—**Cartesian**, **Spherical**, or **Cylindrical**.

- If **Spherical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)** in the table.
- If **Cylindrical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)** and **Center axis direction** values for **x**, **y**, and **z** in each table.

SCALING

Select a **Coordinate stretching type**—**Rational** or **Polynomial**. Select an option from the **Typical wavelength from** list—**Physics interface** or **User defined**. If **User defined** is selected, enter a value or expression for the **Typical wavelength**. The default is 1.

Enter a value or expression for the **PML scaling factor** and the **PML scaling curvature parameter**. The defaults are 1 for both.

Rotated Systems

Use a **Rotated System** ( ) to define rotation about the out-of-plane axis in 2D and Euler angles in 3D.

Model

If you have the MEMS Module, see Gecko Foot: Model Library path **MEMS_Module>Actuators>gecko_foot**.

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

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. Enter a **Out of Plane Rotation** or **Euler Angles**:

In the **Origin of System** field, define 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

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.

3D

For 3D models, enter the **Euler angles (Z-X-Z)** (in radians) in the α, β, and γ fields.

Spherical Systems

Use a **Spherical System** ( ) to define a spherical coordinate system in 3D by its origin, axis direction, and radial base vector direction.

COORDINATE SYSTEM IDENTIFIER

Enter or edit an **Identifier** or use the default identifier name.

SETTINGS

Select a **Frame type**—**Mesh (x,y,z)**, **Material (x,y,z)**, or **Spatial (x,y,z)** (the default), or **Geometry frame coordinate**. The frame type must match the one for the physics interface where the coordinate system is used.

In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `theta`, and `phi`.

Enter the **Origin of system**. The default is an origin coinciding with the one from the global system.

Enter the **Axis direction**. The default axis direction is the z-direction in the global system.

Enter the **Radial base vector direction (θ=π/2, φ =0)** for the angle $\theta = \pi/2$, $\varphi = 0$. The default direction is the x direction in the global Cartesian system.

Pairs

In this section:

About 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:

- An *Identity pair* is a pair that by default make the fields across two connected boundaries (one from each connecting part) 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 interfaces provide special boundary conditions for identity pairs to model “slit conditions” such as resistive layers.

  Use an identity 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.
- A *Contact pair* is a pair that define boundaries where the parts may come into contact but cannot penetrate each other under deformation for modeling of structural contact and multiphysics contact.

  Use a contact pair 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.

Note

Contact pair modeling requires the Structural Mechanics or MEMS Modules.

Identity Pairs

Use an **Identity Pair** node ( ) 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.

Model

- Thin-Layer Diffusion: Model Library path **COMSOL_Multiphysics>Diffusion>thin_layer_diffusion**

GENERAL

Specify the pair name, pair type, and enable manual control of selections for automatically generated pairs.

Pair Name

Enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair.

Pair Type

Set the **Pair type** to change the pair to a contact pair.

Manual Control of Selections

If the pair was created automatically when forming an assembly, the **Manual control of selections** check box is visible. 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.

SOURCE BOUNDARIES

Click the **Activate Selection** button ( ) to define the source boundaries. Select **Manual** or **All Boundaries** for the boundaries on the source 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.

DESTINATION BOUNDARIES

Click the **Activate Selection** button ( ) to define the source boundaries. Select **Manual** or **All Boundaries** for the boundaries on 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.

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.

Contact Pairs

Use a **Contact Pair** node ( ) to specify two selections of boundaries that cannot penetrate each other under deformation. The contact pairs defined boundaries for parts that may 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.

Note

Contact pair modeling requires the Structural Mechanics or MEMS Modules.

Model

If you have the MEMS Module, see Microgripper, Model Library path **MEMS_Module>Piezoelectric_Devices>microgripper**.

If you have the Structural Mechanics Module, see Snap Hook, Model Library path **Structural_Mechanics_Module>Contact_and_Friction>snap_hook**, and Cylinder Roller Contact, Model Library path **Structural_Mechanics_Module>Verification_Models>cylinder_roller_contact**

GENERAL

Specify the pair name, pair type, and enable manual control of selections for automatically generated pairs.

Pair Name

Enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair.

Pair Type

Set the **Pair type** to change the pair to a identity pair.

Manual Control of Selections

If the pair was created automatically when forming an assembly, the **Manual control of selections** check box is visible. 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.

SOURCE BOUNDARIES

Click the **Activate Selection** button (⏻) to define the source boundaries. Select **Manual** or **All Boundaries** for the boundaries on the source 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.

DESTINATION BOUNDARIES

Click the **Activate Selection** button (⏻) to define the source boundaries. Select **Manual** or **All Boundaries** for the boundaries on 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.

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.

Probes

In this section:

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. It is meaningful both to probe while solving as a monitor and diagnostic tool, and to probe after the computation is finished for results analysis. On top of this functionality, also a probe variable with entire model scope is defined. This variable can be used as any other variable, for example in equations, boundary conditions or in 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 4-4 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 **Results** 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 contain the probe data. For further control, specify the table and plot window each probe uses.

When the simulation has finished, click the **Update Results** button ( ) in the probe nodes' **Settings** windows to change the settings for a probe and update the results information. Then right-click the **Definitions** node and select **Update Probes**.

TABLE 4-4: PROBE TYPES AND ICONS BY SPACE DIMENSION

PROBE TYPE	MODEL SPACE DIMENSION	ICON
Domain	3D	
Domain	2D	
Domain	1D	
Boundary	3D	
Boundary	2D	
Boundary	1D	
Edge	3D	
Domain Point	3D	
Domain Point	2D	
Boundary Point	3D	
Global Variable	1D, 2D, and 3D	

Domain Probes, Boundary Probes, and Edge Probes

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, parametric).

Model

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 Structural Mechanics Module:

- For a boundary probe example, see Snap Hook: Model Library path **Structural_Mechanics_Module>Contact_and_Friction>snap_hook**

PROBE SETTINGS

Select an option from the **Type** list—**Average** (the default), **Maximum**, **Minimum**, or **Integral**. If needed, enter or edit a name for the **Probe variable**.

SOURCE SELECTION

From the **Selection** list, select **Manual** or **All domains, All boundaries**, or **All edges**. The default is to use all domains, boundaries, or edges. If **Manual** is selected, select the applicable geometric entity in the **Graphics** window. Select **All domains**, **All boundaries**, or **All edges** to add all the geometric entity levels to the **Selection** box.

EXPRESSION

Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter a user-defined expression in the **Expression** field.

Select any compatible unit for the quantity in the selected expression from the **Unit** list.

Select the **Description** check box to enter a description of the data set (or edit the default).

INTEGRATION 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. When working with multiple frames, select a frame (spatial, material, geometry, or mesh frame) from the **Frame** list for the volume element to be used in the integration.

TABLE AND WINDOW SETTINGS

By default, COMSOL Multiphysics 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.

Point Probe Expressions

A **Point Probe Expression** ( ) can be added as a subnode to **Domain Point Probes** and **Boundary Point Probes**. A default **Point Probe Expression** subnode is always available.

PROBE SETTINGS

Edit or enter a name for the **Probe variable**.

EXPRESSION

Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter a user-defined expression in the **Expression** field.

Select any compatible unit for the quantity in the selected expression from the **Unit** list.

Select the **Description** check box to enter a description of the data set (or edit the default).

TABLE AND WINDOW SETTINGS

By default, COMSOL Multiphysics 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.

Domain Point Probes

Use a **Domain Point Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex-valued number) from a dynamic simulation (a time-dependent, frequency-domain, or parametric simulation). A **Point Probe Expression** ( ) can be added as a subnode to **Domain Point Probes** and **Boundary Point Probes**. A default **Point Probe Expression** subnode is always available.

POINT SELECTION

Select an option from the **Line entry method** list (in 3D)—**Point and surface normal** (the default), **Point and direction**, **Two points**, or **None**. If **Point and surface normal** is selected, 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, as indicated by the value in the **Depth along line** field. If **Point and direction** is selected, 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). If **Two points** is selected, from the **Point being modified** list, first 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. Select **None** to enter coordinates directly.

Enter a **Depth along line** or use the slider to select a value between 0 and 1 that determines the probe location along the line anywhere from the stating point (0) to the ending point (1).

Enter **Coordinates**. For 2D enter **x** and **y** coordinates; for 3D enter **x**, **y**, and **z** coordinates. Select the **Snap to closest boundary** check box to snap the selected points to the grid. In the 3D case, the **Line entry method** switches to **None**.

Also, define an expression for the point probe in the **Point Probe Expression** subnode.

Boundary Point Probes

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). A **Point Probe Expression** ( ) can be added as a subnode to **Domain Point Probes** and **Boundary Point Probes**. A default **Point Probe Expression** subnode is always available.

BOUNDARY SELECTION

In the **Graphics** window, select the boundary to add to the **Boundary Selection** section's **Selection**.

POINT SELECTION

Enter the **Coordinates**. The position of the point is indicated by a red dot on the selected surface in the **Graphics** window. Click on 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.

Also define an expression for the probe in the **Point Probe Expression** subnode.

Global Variable Probes

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).

PROBE SETTINGS

Edit or enter a name for the **Probe variable**.

EXPRESSION

Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**.

Select a **Unit**. Select the **Description** check box to enter a description of the data set (or edit the default).

TABLE AND WINDOW SETTINGS

By default, COMSOL Multiphysics 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.

Selections

In this section:

- About the Selections Node
- Explicit Selections
- Boolean Selections
- Adjacent Selections
- Ball and Box Selections

About the Selections Node

Create selection nodes under model **Definitions** to represent various parts of the geometry and simplify the process of assigning materials, model equations, boundary conditions, and other model features.

Each user-defined Selection can be reused during the model design and renamed with descriptive titles—for example, “Tube,” “Wall,” or “Fluid.” Changes to the Selection (for example, by adding or removing a boundary) updates all features in the model that use that particular Selection.

There are different types of selections—**Explicit Selections**, different types of **Boolean Selections**, selections of adjacent geometric entities (**Adjacent Selections**), and selections by enclosing part of the geometry by a bounding box or ball (**Ball and Box Selections**)—to add these, right-click a **Definitions** node and select from the **Selections** options as listed in Table 4-5.

In the selection nodes' **Settings** windows (except explicit selections) click the **Zoom Selected** ( ) button to zoom in on the selected geometric entities.

TABLE 4-5: SELECTION TYPE DESCRIPTIONS

SELECTION TYPE	ICON	SEE	DESCRIPTION
Explicit		Explicit Selections	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		Boolean Selections	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.
Adjacent		Adjacent Selections	Use the Adjacent node to create the selection as the adjacent geometric entities (boundaries, for example) to one of more selections.
Ball		Ball and Box Selections	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.

See Also

- Object Selection

In the *COMSOL Multiphysics User's Guide*:

- User-Defined Selections
- Creating Selections From Geometric Primitives and Features
- Pasting a Selection From File
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

Explicit Selections

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 that chosen.

Model

- Acoustics of a Muffler: Model Library path: **COMSOL_Multiphysics>Acoustics>automotive_muffler**
- Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics>Diffusion>effective_diffusivity**

INPUT ENTITIES

Select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the geometric entities to add or remove from the selection list.

Select and add geometric entities of the chosen type in the **Graphics** window or other selection methods, or select the **All domains**, **All boundaries**, **All edges**, or **All points** check box to add all of the geometric entities of that type to the list. The selected items are highlighted in the **Graphics** window. Selecting the check box for all geometric entities locks all entities as selected even if the geometry changes.

Important

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output because it can make it invalid by changing the output type, for example, from domains to boundaries.

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.

- 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 model). 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** is the input, the default output is the **Selected boundaries**. Select other options as required. Select **Adjacent edges** or **Adjacent points** to use the edges or points next to the selected boundaries as the selection output (available options depend on the space dimension of the model). Depending on the selection output, choose to include **Exterior edges** (the default) or **Interior edges** for edges, or **Exterior points** (the default) or **Interior points** for points. Click to select or clear the check boxes as needed.
- If **Edge** is the input, the default output is **Selected edges**.

  Select **Adjacent points** to include the points next to the selected edges as the selection output either as **Exterior points** (the default) or **Interior points**. Click to select or clear the check boxes as needed.
- The **Point** output is the same as the selection input (that is, the selected points).

Boolean Selections

Boolean selections 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, for example, a selection is needed that consists of all boundaries except one or a few. Then it is easier to create a selection that is the complement (just the one boundary, for example) and use a **Complement** selection node to create its complement.

The Boolean selection nodes—**Union** ( ), **Intersection** ( ), **Difference** ( ), and **Complement** ( )—contain similar sections.

UNION (INTERSECTION, DIFFERENCE, COMPLEMENT)

Select a **Geometric entity 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.

> ❗ *Important*
>
> If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output because it can make it invalid by changing the output type, for example, from domains to boundaries.

In the **Selections to add** list, for example, add the selections to form the union or other Boolean operations. Click the **Add** button (+) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Add the selections to add (union), intersect, or invert (complement). For the **Difference** selection node add selections to add and selections to subtract. Also delete selections from the lists using the **Delete** button (✕) and move them using the **Move Up** (⇧) and **Move Down** (⇩) buttons.

Adjacent Selections

In addition to explicit selections with some ability to output adjacent geometric entities, form additional selections of adjacent geometric entities based on existing selections using the **Adjacent** ( ) selection type. 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.

ADJACENT

Select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Input selections** list.

In the **Input selections** list, add the selections for which you want to create a selection of adjacent geometric entities. Click the **Add** button (+) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Add the selections for which to form a selection of adjacent entities. Also delete selections from the lists using the **Delete** button (✕) and move them using the **Move Up** (↑) and **Move Down** (↓) buttons.

Important

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output because it can make it invalid by changing the output type, for example, from domains to boundaries.

Under **Output entities**, define the adjacent geometry objects that the selection contains:

- Select **Adjacent domains** to create a selection that contains all domains adjacent to the input selections.
- Select **Adjacent boundaries** to create a selection that contains all boundaries adjacent to the input selections. If the input selections contain domains, select the **Exterior boundaries** and **Interior boundaries** check boxes to control which types of adjacent boundaries the output selection contain. By default, the output selection contains adjacent exterior boundaries only.
- Select **Adjacent edges** to create a selection that contains all edges adjacent to the input selections. If the input selections contain domains or boundaries, select the **Exterior edges** and **Interior edges** check boxes to control which types of adjacent edges the output selection contain. By default, the output selection contains adjacent exterior edges only.
- Select **Adjacent points** to create a selection that contains all points adjacent to the input selections. If the input selections contain domains, boundaries, or edges, select the **Exterior points** and **Interior points** check boxes to control which types of adjacent points the output selection contain. By default, the output selection contains adjacent exterior points only.

Ball and Box Selections

Another way to selection geometric entities is to define an enclosing **Ball** ( ) or **Box** ( ) to select geometric entities that are completely or partially inside the ball or box.

BALL/BOX

Select the level for the geometric entities (domains, boundaries, edges, or points) to select from the **Level** list under **Geometric Entity Level**. In addition, select **From selections** under **Use input selections** to base the selection on other defined selections. In the **Selections** list, add the selections for which you want to create a selection of adjacent geometric entities. Click the **Add** button (✚) to open an **Add** dialog box that contains all existing selections for the selected geometric entity level. Add the selections for which to form a selection using an enclosing ball or box. Also delete selections from the lists using the **Delete** button (✖) and move them using the **Move Up** (⇧) and **Move Down** (⇩) buttons.

For the enclosing ball or box, select the condition for which the selection considers the entities to be enclosed:

- Select **Entity intersects ball** or **Entity intersects box** (the default) to select all geometric entities that intersect the enclosing ball or box; that is, the selection includes all entities that are partially or completely inside the ball or box.
- Select **Entity inside ball** or **Entity inside box** to select all geometric entities that are completely inside the enclosing ball or box; that is, the selection includes all entities that are completely inside the ball or box.
- Select **Some vertex inside box** or **Some vertex inside ball** to select all geometric entities where at least some vertex is inside the enclosing ball or box.
- Select **All vertices inside ball** or **All vertices inside box** to select all geometric entities where all vertices are inside the enclosing ball or box. This selection might differ slightly compared to when selecting **Entity inside ball** or **Entity inside box** if the geometric entity is outside the ball or box at some points between the vertices.

Note

The ball and box selections use the rendering mesh to determine which entities fit the selection condition. Control the detail for the rendering in the **Preferences** dialog box (select **Graphics** and then use the **Detail** list under **Rendering**).

Define the position and dimension of the enclosing box or ball using the fields for the maximum and minimum coordinates for the box or the position and radius for the ball. The inputs are based on the length unit for the geometry.

BALL

Position the center of the ball by entering the center position in the **x**, **y**, and (3D only) **z** fields under **Ball Center** (the unit is the length unit for the geometry). Enter the radius for the ball (disk in 2D) in the **Radius** field under **Ball Radius**. The default is 0 for all these settings.

BOX

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 (3D only) **z minimum** and **z maximum** fields under **Box Limits** (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 of Inf in some of these settings to make the box only partially bounded.

Views

In this section:

The View Nodes

By default a **View** node is added to all 2D ( ), 2D axisymmetric ( ), and 3D ( ) models.

2D 2D Axi	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 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.

3D

For 3D models, a **Camera** subnode ( ) and three **Directional Light** nodes ( ), including default settings, are also added.

The **View** nodes and 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 (directional lights, headlights, spotlights, and point lights), lighting attributes (diffuse, specular, and ambient light), and camera settings.

Also right-click the **View** node to add **Hide Geometric Entities** and **Hide Geometry Object** nodes for any space dimension.

If a **View** is customized—for example, by renaming it *Transparency with spotlight*—it can be selected from the list of views ( ) on the **Graphics** toolbar. Create additional **View** nodes by right-clicking **Definitions** ( ) and selecting **View** and then experiment by switching views to find the best way to illustrate a model.

Note

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.

To reset the **View** node's settings and its subnodes to the default settings, right-click the **View** node and select **Reset to Default** ( ).

See Also

- User-Defined Views in the *COMSOL Multiphysics User's Guide*

The 2D View Node

2D

2D Axi

For 2D and 2D axisymmetric models, the **View** node (xy) controls the settings to display or hide geometry labels and lock the axis.

Also right-click the **View** node to add **Hide Geometric Entities** and **Hide Geometry Object** nodes.

VIEW

Select the **Show geometry labels** check box to display the geometry short name label in the **Graphics** window. The labels appear for geometry objects or geometric entities (domain labels, boundary labels, edge labels, or point labels), depending on what selections you are making.

Select the **Show direction arrows** check box to display the direction arrows on boundaries 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.

See Also

- The View Nodes
- The 3D View Node
- Hide Geometry Objects and Hide Geometric Entities

Axis

2D

2D Axi

For 2D and 2D axisymmetric models, the **View** node has an **Axis** (yx) node where you can set the axis coordinates, define manual spacing, and preserve the aspect ratio.

AXIS

Enter **x minimum** and **y minimum** and **x maximum** 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. 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

Enter **Extra x** and **Extra y** values. 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. Click the **Range** button ( ) as required.

See Also

- The View Nodes
- Entering Ranges and Vector-Valued Expressions
- Hide Geometry Objects and Hide Geometric Entities

The 3D View Node

The 3D **View** node ( ) 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 short name in the **Graphics** window.
- Select the **Show 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 by revisiting the **View** node restore the camera settings s to the values in the view at the time the **Lock camera** check box is 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 for the **Ambient intensity** 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.

See Also

- The View Nodes
- About 3D User-Defined Views in the *COMSOL Multiphysics User's Guide*

Camera

Use the **Camera** node ( ) to orient the camera view in 3D models.

I

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** 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.

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, or enter **x, y**, and **z** coordinates for the camera target location.

UP VECTOR

In the **Graphics** window, 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.

CENTER OF ROTATION

By default the center of rotation is 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 or, enter **x** and **y** coordinates. If the view's camera setting uses orthotropic projection, zoom in or out by increasing or decreasing the value in the **Orthographic scale** field. For a perspective projection this setting has no effect.

See Also

- The View Nodes
- The 3D View Node

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. The direction of the light, the light and specular intensity levels, and the color can be adjusted as required.

DIRECTION

Enter **x**, **y**, and **z** coordinates to place the arrow at the coordinates where the light is to hit the object. The arrow points in the direction of the light.

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**. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**.
- 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.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light (directional, point, spot, or headlight) being applied to the object. Click to clear the check box as required and remove the marker from the Graphics window display.

See Also

- The View Nodes
- The 3D View Node
- About the 3D View Light Sources and Attributes in the *COMSOL Multiphysics User's Guide*

Point Light

The **Point Light** ( ) has a position and emits the light equally in all directions; it is like a light bulb. The position of the light, the light and specular intensity levels, and the color can be adjusted as required.

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**. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**.
- 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.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light (directional, point, spot, or headlight) being applied to the object. Click to clear the check box as required and remove the marker from the Graphics window display.

See Also

- The View Nodes
- The 3D View Node
- About the 3D View Light Sources and Attributes in the *COMSOL Multiphysics User's Guide*

Spotlight

The **Spotlight** ( ) node acts like a flashlight and has both a position and a direction. Set the position and direction of the light, the light and specular intensity levels, and the color as required. In addition, the spread angle can be adjusted. The width of the light marker corresponds to the spread angle.

POSITION

Enter **x**, **y**, and **z** coordinates for the position of the spotlight.

DIRECTION

Enter **x**, **y**, and **z** coordinates for the direction of the light from the spotlight.

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.
- 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**. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**.
- 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.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light (directional, point, spot, or headlight) being applied to the object.

Click to clear the check box as required and remove the marker from the Graphics window display.

See Also

- The View Nodes
- The 3D View Node
- About the 3D View Light Sources and Attributes in the *COMSOL Multiphysics User's Guide*

Headlight

A headlight is a directional light, which points to the scene from the camera position. The **Headlight** ( ) node'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.

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**. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light (directional, point, spot, or headlight) being applied to the object.

Click to clear the check box as required and remove the marker from the Graphics window display.

See Also

- The View Nodes
- The 3D View Node
- About the 3D View Light Sources and Attributes in the *COMSOL Multiphysics User's Guide*

Hide Geometry Objects and Hide Geometric Entities

Hide a set of geometry objects—in the created or imported geometry—or geometric entities—in the analyzed (finalized) geometry used for modeling—at a specific geometric entity level for any View.

GEOMETRIC SCOPE

- Hide a set of geometry objects at a specific **Geometric entity level**—**Object**, **Domain**, **Boundary**, **Edge** (3D only), or **Point**.
- Hide a set of geometric entities at a specific **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point**.

See Also

- The View Nodes
- The 2D View Node
- The 3D View Node

5

Solver Features

This chapter describes the solver features available in COMSOL Multiphysics. For more details and references for the solver algorithms, see Advanced Solver Topics.

The Study Nodes

Study

A **Study** node holds the nodes that define how to solve a model. These nodes are divided into three broad categories:

- Study steps, which determines overall settings suitable for a certain study type.
- Solver configurations, which contains the solvers and related configurations for dependent variables to solve for, intermediate storage of solutions, and specific solver settings. It is only visible if it has contents or you click the **Show** button ( ) and select **Advanced Study Options**.
- Job configurations (distributed parametric jobs, batch jobs, and cluster computing), which is only visible if it has contents or you click the **Show** button ( ) and select **Advanced Study Options**.

The topics are described in the following sections. The main **Study** nodes contain 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 interfaces in the study are generated automatically when computing the solution. Clear the **Generate default plots** check box if you do not want any default plots to be generated.

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 the **Generate convergence plots** check box if you do not want convergence plots to be generated during the solution process.

ADDING A STUDY

To add a new study to a model, right-click the root node in the model tree and select **Add Study** ( ). You can then select preset study types and other study types on the **Select Study Type** page in the **Model Wizard**.

Study Types

BASIC STUDY TYPES

COMSOL Multiphysics has four basic study types:

- Stationary
- Time Dependent, Time Discrete, and Time-Dependent Modal
- Eigenfrequency or Eigenvalue
- Frequency Domain or Frequency Domain Modal

PARAMETRIC SWEEPS, MULTIGRID LEVELS, BATCH, AND CLUSTER COMPUTING NODES

In addition, Parametric Sweep nodes can be added to a study to perform parametric variation on the other studies, or a Multigrid Level subnode to the other study step nodes to describe a geometric multigrid level used by the study. Also add Batch and Cluster Computing nodes as required. Some additional application-oriented study types are only included with some of the add-on modules.

ADDITIONAL STUDY TYPES

There are also a variety of studies that require specific Module licenses or interfaces. These studies are sometimes based on one of the four basic study types (for example, Frequency-Domain, Perturbation) or are specific to the interface (for example, Mean Energies). These studies are also loosely grouped by type:

Time Dependent

- Time-Dependent with Initialization, Fixed Geometry
- Time-Dependent with Initialization
- Transient with Initialization
- Frequency-Transient

Stationary

- Stationary Plug Flow
- Stationary with Initialization
- Frequency-Stationary

Study Steps

- Wall Distance Initialization

- Time-Dependent, Fixed Geometry
- Current Distribution Initialization

Harmonic Perturbation, Prestressed Analysis, or Signal Analysis

- Frequency-Domain, Perturbation
- Small-Signal Analysis, Frequency Domain
- Prestressed Analysis, Eigenfrequency
- Prestressed Analysis, Frequency Domain
- AC Impedance Stationary
- AC Impedance Time Dependent

Structural Mechanics Module

- Linear Buckling

Acoustics Module

- Mode Analysis

RF Module

- Boundary Mode Analysis

Plasma Module

- Mean Energies

Particle Tracing Module

- Particle Trajectories Study

See Also

- The Study Node Settings Window
- Solver Studies and Study Types and Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*
- Solver API in the *COMSOL Java API Reference Guide*

The Study Node Settings Window

Add a study to your model by right-clicking a **Study** node and then selecting the study type from the menu. If you select a study for your physics interfaces in the **Model Wizard**, a study feature appears under the **Study** node that is added to the model when you click **Finish**. The **Settings** window for all study types contains the following section:

VALUES OF DEPENDENT VARIABLES

The settings in this section determine how the solver handles dependent variables that you do not solve for. This is applicable in a solver sequence where you only solve for a subset of the dependent variables in each step.

To specify the values of dependent variables that you do not solve for, select the **Values of variables that you do not solve 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 interfaces in the model.
- **Solution** to use initial values as specified by a solution object.

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 **Stationary** list, select **Automatic** (the default) to use the last (typically the only) solution, **All** to use all (typically just one) solution from that study, or **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 **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, or select one of the output times to use the solution at that time.
- For an eigenvalue study, from the **Eigenvalue** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **All** to use all solutions from that study, or select one of the eigenvalues to use the corresponding eigensolution.

See Also

- Solvers and Study Types in the *COMSOL Multiphysics User's Guide*
- Solver API in the *COMSOL Java API Reference Guide*

Stationary

A **Stationary** study node ( ) corresponds to a stationary solver (default), a parametric solver, or an optimization solver. The Stationary study generates a solver that is used to solve a stationary problem.

Use the *parametric continuation solver* (which you activate by the **Continuation** check box in the **Extension** section) 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.

RESULTS 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 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.

MESH SELECTION

Select a mesh to use for each geometry in the study. 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.

PHYSICS SELECTION

The **Physics interface** column contains the names of all physics interfaces in the model. You can choose to not include one or more of the physics interfaces by clicking the ✓ button in the **Use** column (all physics interfaces are included by default). The ✓ button then changes to an ✗ to indicate that the physics interface is not included. Click the ✗ button to include the physics interface again. 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 interface node's settings. Changing it affects the discretization order used by this study. To add another discretization, use a **Discretization** node in the physics interface.

EXTENSION

This section contains some extensions for the study, such as a parametric continuation solver or adaptive mesh refinement.

To use the parametric continuation solver, select the **Continuation** check box. Specify the parameter to vary in the **Continuation parameter** field. The parameter can occur in equations, constraints, or expressions but not as a geometry or mesh parameter, for example. The list you enter in the **Parameter values** field is a vector. It must contain a monotonically increasing or decreasing sequence of real parameter values, for example, `range(0,0.1,1.5)` or `0 10^(range(-3,2))`.

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.

Tip

Select the **Optimization** check box if the study is to be used for optimization purposes. The optimization solver requires an Optimization Module license.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Time Dependent

The **Time Dependent** study node ( ) gives a solver with a **Time-Dependent Solver** node ( ) or an **Optimization Solver** node ( ).

STUDY SETTINGS

Specify the time interval for the output from the simulation in the **Times** field. 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 gives time steps from 0 to 1.5 s with a step size of 0.1 s; or use any combinations of such input. Check the **Relative tolerance** setting if you want 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.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

EXTENSION

Note

This section contains some extensions for the study. The options are mutually exclusive and only one of the check boxes can be selected.

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.

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.

Tip

Select the **Optimization** check box if the study is to be used for optimization purposes. The optimization solver requires an Optimization Module license.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Time Discrete

Note

The settings available for the **Time Discrete** node ( ) are some of the settings available for the Time Dependent node.

The time-discrete solver is used to solve your fluid dynamics problem using a projection method. Also, problems which have been discretized in time using the `prev` or `bdf` operator are solved with the time-discrete solver.

Time-Dependent Modal

Note

The settings for the **Time-Dependent Modal** study node ( ) are the same as those for the Time Dependent node.

Eigenfrequency

Use an **Eigenfrequency** study node ( ) when you want to solve an eigenvalue problem for a set of eigenmodes and associated eigenfrequencies. An Eigenfrequency study node gives a solver with an **Eigenvalue Solver** node ( ).

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.

Note

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.

MESH SELECTION

Select which 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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

EXTENSION

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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Eigenvalue

Use an **Eigenvalue** study node ( ) when you want to solve an eigenvalue problem for a set of eigenmodes and associated eigenvalues. An Eigenvalue study node gives a solver sequence with an **Eigenvalue Solver** node ( ).

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.

> Note
>
> 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.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

EXTENSION

Select the **Adaptive mesh refinement** check box if you wish to use adaption. Select the geometry to use for the adaption from the **Adaption in geometry** list.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Frequency Domain

The **Frequency Domain** node ( ) corresponds to a frequency sweep.

EXTENSION

In addition to the frequency sweep, you can merge it with an auxiliary sweep over one or several extra parameters. Select the **Auxiliary sweep** check box to enable the extra parameter sweep. The final sweep is a multiparameter sweep with the frequency as the last parameter; see Parametric Sweep for more information about multiparameter sweeps.

You enable the *asymptotic waveform evaluation (AWE) solver* by selecting the **Use asymptotic waveform evaluation** check box. The Frequency Domain study generates a solver sequence that is used to solve a stationary parametric problem or an Asymptotic Waveform Evaluation problem.

RESULTS 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 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.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Frequency Domain Modal

The **Frequency Domain Modal** study node ( ) corresponds to a modal frequency sweep for systems with frequency-based loads.

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.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Multigrid Level

Click the **Show** button ( ) and select **Advanced Study Options** and then right-click any of the **Study Step** nodes to add a **Multigrid Level** node ( ). This is a subnode to the other study step nodes and it describes a geometric multigrid level used by the study.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

Parametric Sweep

Use a **Parametric Sweep** study node ( ) 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 (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), you can add several **Parametric Sweep** nodes, one for each parameter. 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.

STUDY SETTINGS

Specify the parameters to vary in the **Parameter names** field. Each parameter must be globally defined in the model tree. As usual, you add parameters to the field by clicking the plus sign (+) and select from the appearing list of parameter names.

The parameters can occur in equations, constraints, or expressions, including geometry and mesh settings. The list you enter in the **Parameter values** field is a vector. If you are varying one parameter, the parameter must contain a monotonically increasing or decreasing sequence of real parameter values, for example, `range(0,0.1,1.5)` or `0 10^(range(-3,2))`.

When solving for several parameters, the list must be monotonic in the first parameter. Enter the list as a vector where the parameter sets are stored consecutively. For example, if there are two solver parameters, the entry `0 5 2 3 4 8` in the **Parameter values** field means that the parametric solver is called with the parameter/value pairs $(0, 5)$, $(2, 3)$, and finally $(4, 8)$. For three solver parameters, the same list would correspond to the parameter sets $(0, 5, 2)$ and $(3, 4, 8)$.

To avoid the constraint that the list must be monotonic in the first parameter, you can use a dummy variable as the first parameter. If you want to use the parameter-value pairs $(7, 8)$, $(4, 12)$, and $(1, 6)$, for example, add an unused dummy parameter, say `p0`, first in the **Parameter names** field. Then use the entry `1 7 8 2 4 12 3 1 6` in the **Parameter values** field. The parametric solver then uses the parameter sets $(1, 7, 8)$, $(2, 4, 12)$, and $(3, 1, 6)$, where the first value in each set applies to the first parameter, `p0`, which is a dummy parameter that does not appear in the model settings.

Note

For a full (or filled) multiparameter sweep, use a separate **Parametric Sweep** node for each parameter. The program then loops over all combinations of parameter values.

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 **Parameter values** field.

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.

Note

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.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, etc.) 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 will be used.

Note

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 manipulate the table data and make so called response surfaces directly from the **Results** view toolbar **Surface Plot** button

The **Memory conservation settings for jobs** and the **Keep solutions in memory** selector can be used to control how to store the solutions from the individual parametric sweep solutions. Select **All** to store all the parametric sweep solutions in memory and 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 will store the separate parametric sweep solutions and its corresponding model in separate mph-files.

Batch

Use a batch job to start a COMSOL batch process that solves the current study on your computer. Click the **Show** button ( ) and select **Advanced Study Options** and then right-click the **Study** node to add a **Batch** study node ( ). Once you have set the filename and directory, right-click the parent Study node and choose **Compute** to start a COMSOL batch process that computes the current study. When the process has started the COMSOL Desktop enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Stop Logging** button in the **Progress** window. When the process finishes or you click the **Stop Logging** button, an External Process node that represents the current running process appears under a **Batch Data** node, which in turn appears as a subnode of the Batch node under the **Job Configurations** node ( ). For more information, see Batch, Batch Data, and External Process.

Important

You cannot create a Batch study and a Cluster Computing study in the same study at the same time.

STUDY 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.

Cluster Computing

Use the **Cluster Computing** study node ( ) when you want to submit COMSOL batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster. For more information, read about the job feature Cluster Computing and its related features. 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.

> *Important*
>
> You cannot create a Batch study and a Cluster Computing study in the same study at the same time.

Click the **Show** button ( ) and select **Advanced Study Options** and then right-click the **Study** node to add a **Cluster Computing** study node ( ) to run a batch job 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. Once the process is started the COMSOL Desktop enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Stop Logging** button in the **Progress** window. Once the process is finished or you click the **Stop Logging** button, an **External Process** node is selected that represents the current running process. Click the **Save as Default** button ( ) in the **Settings** window's toolbar to save the current setting as default. 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.

STUDY SETTINGS

You choose which type of cluster job you want to do from the **Cluster type** list:

- The **General** type (the default setting) can be configured to run on many types of clusters and schedulers, including Linux clusters.
- 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.
- The **Not distributed** choice is useful when you want to submit a batch job to a job scheduler without running a distributed job.

If you are using the **General** type and you have started a multiprocessor daemon (MPD) on the computer 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.

If your cluster is Linux and it requires ssh (secure shell) or rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol.

You set the number of compute nodes (physical nodes) to use in the **Number of nodes** field.

In the **WCCS 2003** and **HPCS 2008** cases you can specify the **User** and the name of the scheduler host in **Scheduler**:

- 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.
- 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`.

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 typing it directly or clicking the **Browse** button to choose a batch directory.

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 by browsing by first clicking **Browse**. The cluster job uses this path from the compute node to access the input file and write back the result. 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 type the path to the directory in the **Directory** field). This can occur if you are submitting jobs to a job scheduler. Typically 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\COMSOL42`.

Use the **Save as Default** (💾) button in the toolbar to save the current directory settings as the default settings.

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.

Time-Dependent with Initialization, Fixed Geometry

> Note
>
> The **Time-Dependent with Initialization, Fixed Geometry** study is available with the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces, which require the Electrodeposition Module.

The **Time-Dependent with Initialization, Fixed Geometry** 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.

Time-Dependent with Initialization

> Note
>
> The **Time-Dependent with Initialization** study is available with the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces, which require the Electrodeposition Module.

The **Time-Dependent with Initialization** study ( ) adds a Current Distribution Initialization study step and Time Dependent study step to the **Study** node. See the study steps for settings information.

Time-Dependent, Fixed Geometry

> Note
>
> The **Time-Dependent, Fixed Geometry** study step is added as a study step to the Time-Dependent with Initialization, Fixed Geometry, which is available with the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces. These interfaces require the Electrodeposition Module.
>
> The settings available for this study step are described for the Time Dependent node.

Use the **Time-Dependent, Fixed Geometry** study step ( ) 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.

Current Distribution Initialization

> Note
>
> The **Current Distribution Initialization** study step is available with the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces, which require the Electrodeposition Module.
>
> The settings available for this study step are described for the Time Dependent node.

The **Current Distribution Initialization** study step ( ) is added to the Time-Dependent with Initialization, Fixed Geometry and Time-Dependent with Initialization studies. It can be used 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 of the problem.

Stationary Plug Flow

> Note
>
> The **Stationary Plug Flow** study is available with the **Reaction Engineering** interface, which requires the Chemical Reaction Engineering Module.
>
> The settings available for this study are described for the Time Dependent node.

The **Stationary Plug Flow** study node ( ) is used to solve the equations set up by the predefined Plug-flow reactor type. The reactor equations describe the molar flow rates (SI unit: mol/s) as function of reactor volume (SI unit: m^3) under stationary conditions. Selecting the Reaction Engineering interface and the Stationary Plug Flow study in the Model Wizard sets up a Plug-flow reactor type by default.

Frequency-Domain, Perturbation

Note

The settings for the **Frequency-Domain, Perturbation** node ( ) are the same as those for the Frequency Domain node.

Use perturbed frequency domain analysis to study small oscillations about a bias solution (small-signal analysis).

Note

The **Frequency-Domain, Perturbation** 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 either the Batteries & Fuel Cells Module or the Electrodeposition Module.
- As a study step for the Small-Signal Analysis, Frequency Domain study, which requires either the AC/DC Module or MEMS Module.
- As a study and study step with the **Thin-Film Flow** and **Lubrication Shell** interfaces, which require the CFD Module or MEMS Module.
- As a study and study step with the **Film Damping Shell** and **Thin-Film Gas Flow** interfaces, which require the MEMS Module.

With the **Frequency-Domain, Perturbation** 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 from the Model Wizard, two study steps are added under the Study node. When it is added as a study step (for example, with the AC Impedance study types), it is also part of a two step study that is added under the Study node.

In all cases, 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.

See Also

In the *COMSOL Multiphysics User's Guide*:

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis
- Physics Interface Exclusive and Contributing Feature Node Types

Small-Signal Analysis, Frequency Domain

Note

The **Small-Signal Analysis, Frequency Domain** study is available with the:

- **Electric Currents** and **Magnetic Fields** interfaces, which both require either the AC/DC Module or MEMS Module, and
- **Electric Currents-Shell** and **Magnetic and Electric Fields** interfaces, which requires the AC/DC Module.

The settings available for this study are described for each study step.

Use the **Small-Signal Analysis, Frequency Domain** study ( ) for perturbed frequency domain studies of small oscillation about a bias solution. The study creates two study steps, where the first (a Stationary study node) solves for a stationary problem, and the second step (a Frequency-Domain, Perturbation study node) solves for a harmonic perturbation in the frequency domain of the stationary solution.

See Also

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*.

Prestressed Analysis, Eigenfrequency

Note

The **Prestressed Analysis, Eigenfrequency** study is available for the:

- **Solid Mechanics** interface using the Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module,
- **Shell**, **Plate**, and **Truss** interfaces using the Structural Mechanics Module, and
- **Electromechanics** interface using the MEMS Module.

Use the **Prestressed Analysis, Eigenfrequency** study ( ) to compute eigenfrequencies which are influenced by a prior static load. It is necessary to have a geometric nonlinearity active in order to perform this type of analysis. Adds a Stationary and an Eigenfrequency study step to the Study node. The settings are described for the study steps.

See Also

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*

Prestressed Analysis, Frequency Domain

Note

The **Prestressed Analysis, Frequency Domain** study is available for the:

- **Solid Mechanics** interface using the Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module, and the
- **Electromechanics** interface using the MEMS Module.

Use the **Prestressed Analysis, Frequency Domain** study ( ) to compute the response to harmonic loads fluctuating around a static pre-load. It is necessary to have a geometric nonlinearity active in order to take the influence on the stiffness from the pre-load into account. Adds a Stationary and a Frequency Domain study step to the Study node. See the study steps for settings.

See Also

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*

AC Impedance Stationary

Note

The **AC Impedance Stationary** study is available with the:

- **Battery with Binary Electrolyte**, **Lead Acid Battery**, and **Lithium-Ion Battery** interfaces which require the Batteries & Fuel Cells Module,
- **Secondary Current Distribution** and **Tertiary Current Distribution, Nernst-Planck** interfaces, which are available for both the Batteries & Fuel Cells Module and the Electrodeposition Module.

The **AC Impedance Stationary** study node () creates two study steps, where the first (a Stationary study node) solves for a stationary problem, and the second step (a Frequency-Domain, Perturbation study node) solves for a harmonic perturbation in the frequency domain of the stationary solution. See the study steps for settings.

See Also

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*.

AC Impedance Time Dependent

Note

The **AC Impedance Time Dependent** study is available with the:

- **Battery with Binary Electrolyte**, **Lead Acid Battery**, and **Lithium-Ion Battery** interfaces which require the Batteries & Fuel Cells Module,
- **Secondary Current Distribution** and **Tertiary Current Distribution, Nernst-Planck** interfaces, which are available for both the Batteries & Fuel Cells Module and the Electrodeposition Module.

The **AC Impedance Time Dependent** study node ( ) creates two study steps, where the first (a Time Dependent study node) solves for a time-dependent problem, and the second step (a Frequency-Domain, Perturbation study node) solves for a harmonic perturbation in the frequency domain of the solution at the last time step. See the study steps for settings.

See Also

- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis in the *COMSOL Multiphysics User's Guide*.

Linear Buckling

Note

The **Linear Buckling** study is available for the **Solid Mechanics** interface using the Structural Mechanics Module or the MEMS Module.

When you add the **Linear Buckling** study ( ), two study steps are added. The first is a Stationary study step, and then followed by a **Linear Buckling** step as the second step. In this step, an eigenvalue solver computes the buckling modes and the associated critical load factors.

STUDY SETTINGS

Use the **Desired number of buckling modes** field to specify the number of buckling modes you want the eigenvalue solver to return.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Mode Analysis

Note

The **Mode Analysis** study is available with the **Boundary Mode Acoustics** and **Boundary Mode Aeroacoustics** interfaces which require the Acoustics Module.

A **Mode Analysis** study node ( ) (and study step) computes the modes for an acoustics or electromagnetic wave using an eigenvalue solver. When this study is added, it adds a Mode Analysis study step under the Study node.

STUDY SETTINGS

Enter a **Desired number of modes**. The default is 6.

Enter a value or expression in the **Search for modes around** field. The default is 0.

Select a method to **Transform**—**Out-of-plane wave number** (the default) or **None**.

Enter a value or expression for the **Mode analysis frequency**. The default is 100 Hz.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

EXTENSION

Note

This section contains some extensions for the study. The options are mutually exclusive and only one of the check boxes can be selected.

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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Stationary with Initialization

Note

The **Stationary with Initialization** study is available with these interfaces, which require either a CFD Module or Heat Transfer Module:

- All types of **Turbulent Flow, Low Re k-**ε interfaces (Conjugate Heat Transfer and Non-Isothermal Flow).

With the CFD Module, these interfaces also have this study available:

- All types of **Turbulent Flow, Spalart-Allmaras** interfaces (Conjugate Heat Transfer, Non-Isothermal Flow, and High Mach Number Flow).

The **Stationary with Initialization** study ( ) is for stationary two-phase flow models that require an initialization of a level set function or phase field function. It adds a Wall Distance Initialization study step followed by a Stationary study step to the Study node. The first step, the **Wall Distance Initialization** study step, is dedicated to solve for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface. The second step is an ordinary **Stationary** study step, but it excludes the reciprocal wall distance except for turbulent flow interfaces where, if the interface is put on a moving frame, the reciprocal wall distance is solved for in the stationary step as well.

Transient with Initialization

Note

The **Transient with Initialization** study is available with these interfaces, which require either a CFD Module or Microfluidics Module:

- **Laminar Two-Phase Flow** and **Turbulent Two-Phase Flow, Level Set**
- **Laminar Two-Phase Flow** and **Turbulent Two-Phase Flow, Phase Field**
- **Mathematics Moving Interface>Level Set**

With either the CFD Module or Heat Transfer Module, these interfaces also have this study available:

- All types of **Turbulent Flow, Low Re k-ε** interfaces (Conjugate Heat Transfer and Non-Isothermal Flow).

With the CFD Module, these interfaces also have this study available:

- All types of **Turbulent Flow, Spalart-Allmaras** interfaces (Conjugate Heat Transfer, Non-Isothermal Flow, and High Mach Number Flow).

The **Transient with Initialization** study ( ) is for time dependent two-phase flow models that require an initialization of a level set function or phase field function. It adds a Wall Distance Initialization study step followed by a Time Dependent study step to the Study node. The first step, the **Wall Distance Initialization** study step, is dedicated to solve for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface. The second step is an ordinary **Time Dependent** study step.

Wall Distance Initialization

Note

See the Stationary with Initialization and Transient with Initialization studies for availability by Module.

The settings for this study are the same as for the Stationary and Time Dependent studies.

The **Wall Distance Initialization** study step is added to the Stationary with Initialization and Transient with Initialization studies. This first step is dedicated to solve for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface.

Boundary Mode Analysis

Note

The **Boundary Mode Analysis** study is available with the **Electromagnetic Waves** or **Microwave Heating** interfaces, which both require the RF Module.

A **Boundary Mode Analysis** study node ( ) combines a mode analysis on a port (boundary) (which can represent, for example, a cross section of a waveguide) with a frequency domain study for the full geometry. Adds a **Boundary Mode Analysis** study step followed by a Frequency Domain study step to the Study node.

STUDY SETTINGS

Enter a **Desired number of modes**. The default is 1.

Enter a value or expression in the **Search for modes around** field. The default is 1.

Select a method to **Transform**—**Effective mode index** (the default), **Out-of-plane wave number**, or **None**.

Enter a **Port name**. The default is 1.

Enter a value or expression for the **Mode analysis frequency**. The default is 1e9 Hz.

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.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics interfaces 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.

EXTENSION

Note

This section contains some extensions for the study. The options are mutually exclusive and only one of the check boxes can be selected.

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.

VALUES OF DEPENDENT VALUES

See The Study Node Settings Window for details.

Frequency-Stationary

Note

The **Frequency-Stationary** study is available with the:

- **Induction Heating** interface, which requires the AC/DC Module.
- **Microwave Heating** interface, which requires the RF Module.

See Stationary for all the settings.

Frequency-Transient

Note

The **Frequency-Transient** study is available with the:

- **Induction Heating** interface, which requires the AC/DC Module.
- **Microwave Heating** interface, which requires the RF Module.
- **Inductively Coupled Plasma** interface, which requires both the AC/DC Module and the Plasma Module.
- **Microwave Plasma** interface, which requires both the RF Module and the Plasma Module.

Note

See Time Dependent for all the settings.

Physics interfaces that support this study compute electromagnetic fields in the frequency domain and the temperature (electron temperature in the case of the **Inductively Coupled Plasma** and **Microwave Plasma** interfaces) in the time domain.

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.

Mean Energies

Note

The **Mean Energies** study is available with the **Boltzmann Equation, Two-Term Approximation** interface, which requires the Plasma Module.

Use the **Mean Energies** study ( ) to enter an array of values for the mean electron energy. COMSOL will compute 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.

Particle Trajectories Study

Note

The **Particle Trajectories** study is available with particle interfaces, which requires the Particle Trajectories Module.

Use the **Particle Trajectories** study ( ) and study step has the same settings as the Time Dependent study step except that by default, only the Particle Tracing Module interfaces are active in the physics selection.

Solver Configurations

In this section:

- The Solver Configurations Node
- Show Default Solver
- Adding Solver Nodes
- Running a Solver Configuration
- Errors and Warnings

See Also

- Solvers and Study Types in the *COMSOL Multiphysics User's Guide*
- Solver API in the *COMSOL Java API Reference Guide*

The Solver Configurations Node

The **Solver Configurations** node ( ) contains all solver sequences defined for a study. The node's context menu has these options. Select:

- **Show Default Solver** ( ) to show the node that corresponds to the study step nodes.
- **Clear Solutions** ( ) to remove any solutions in the solver.
- **Delete Solvers** ( ) to delete all solvers under the **Solver Configurations** node.
- From the main menu, click the **Show** button ( ) and select **Advanced Study Options** and then right-click the **Solver Configurations** node to add a **Create Custom Solver** ( ) to add a **Solver** node without any added solver settings or other nodes.

See Also

- Clearing, Deleting, Disabling, and Enabling a Solver or Solutions in the *COMSOL Multiphysics User's Guide*

Show Default Solver

To display the solver that corresponds to the study steps included in a study and the current physics settings, right-click the **Study** node and select **Show Default Solver** ( ) from the context menu. Or right-click the **Solver Configurations** node and select **Show Default Solver**.

Adding Solver Nodes

To add solver nodes to a solver configuration, right-click the **Solver Configurations** node and then select from the context menu.

Running a Solver Configuration

Running a *solver configuration* is tantamount to computing a solution. Like mesh nodes, solution nodes are not built automatically as they are added. To compute a solution, right-click the corresponding solver node and select **Compute**. Or right-click a study node and select **Compute** to compute the enabled solver sequence with a green border around its icon, if such a solver sequence exist. If no solver sequence exists or if all sequences are disabled, a new solver sequence is generated and computed. If you have already generated a solver sequence for the study, you can compute some or all subnodes in it by selecting **Compute to Selected** ( ), **Compute from Selected** ( ), or **Compute All** ( ). For example, right-click a **Dependent Variables** node and select **Compute to Selected** ( ) to evaluate the initial values for the dependent variables.

Errors and Warnings

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 **Problems** node (⚠).

Solver Overview

In this section:

- Solver Configurations
- Solver Features Overview

Solver Configurations

You solve a model by computing a *solver configuration*, which is a scheme for computing a solution. A **Solver Configuration** node displays if it has content or if the the **Show** button ( ) is clicked and **Advanced Study Options** selected to add this node to the **Model Builder**. A model can also be solved by computing a study, which defines a sequence of solver configurations and, in some cases, a sequence of job configurations.

Loosely speaking a solver configuration consists of one or more **Solver** nodes ( ). Each Solver node consists of a sequence of subnodes specifying how to compute the solution. Typically, such a solver sequence contains information about which physics and geometry to use, which variables to solve for, and which type of study to perform.

Solver Features Overview

There are three types of features that appear as subnodes to a Solver node: *operation features*, *attribute features*, and *utility features*. Operation features produce solutions as output. In particular, the output from the last executed operation feature is available for results analysis and visualization. Attribute features hold properties that control the behavior of operation features. Utility features handle special types of operations. Some of the settings in subnodes are in sync with the corresponding Study setting. They are greyed out by default and can only be controlled from the subnode by changing the **Defined by study step** setting to **User defined**.

See Also

- Solution Operation Features
- Solution Attribute Features
- Solution Utility Features

AVAILABLE SOLUTION OPERATION FEATURES

Table 5-1 lists the available Solution Operation Features.

TABLE 5-1: SOLUTION OPERATION FEATURES

FEATURE	ICON	PURPOSE
AWE Solver		Solve a parametric problem with asymptotic waveform evaluation.
Eigenvalue Solver		Solve an eigenvalue problem.
Modal Solver		Solve time-dependent or parametric problem with modal analysis.
Optimization Solver		Optimization parameters (requires the Optimization Module).
Plugflow Solver		Solve a plug-flow reactor model (requires the Chemical Reaction Engineering Module).
Stationary Solver		Solve a stationary problem.
Time-Dependent Solver		Solve a time-dependent problem using an implicit time stepping scheme.
Time Discrete Solver		Solve a time-dependent problem which has been discretized in time.
Time Explicit Solver		Solve a time-dependent problem using an explicit time-stepping scheme.
Dependent Variables	u,v,w	Handles the dependent variables solved for (initial values, scaling) and dependent variables not solved for (prescribed values).

AVAILABLE SOLUTION ATTRIBUTE FEATURES

Table 5-2 lists the available Solution Attribute Features.

TABLE 5-2: SOLUTION ATTRIBUTE FEATURES

FEATURE	ICON	PURPOSE
Adaptive Mesh Refinement		Adaptive mesh refinement parameters.
Advanced		Advanced general parameters.
Automatic Remeshing		Automatic remeshing parameters.
Coarse Solver		Handles settings for the coarse solver (when using Multigrid).
Direct		Handles settings for a direct linear solver.

TABLE 5-2: SOLUTION ATTRIBUTE FEATURES

FEATURE	ICON	PURPOSE
Field		Handles settings for field variables. Each field variable needs a separate Field node. This attribute is used together with the Dependent Variables operation feature.
Fully Coupled		Handles parameters for a fully coupled solution approach.
Incomplete LU		Handles settings for an incomplete LU solver or preconditioner.
Iterative		Handles settings for an iterative linear solver or preconditioner.
Jacobi		Handles settings for a Jacobi linear solver or preconditioner.
Krylov Preconditioner		Handles settings for a Krylov-type linear solver or preconditioner.
Lower Limit		Parameters for imposing restrictions of degrees of freedom.
Lumped Step		Available when using the segregated attribute feature. 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.
Parametric		Handles settings for parameter stepping. This attribute can be used together with the Stationary Solver.
Postsmoother		Handles settings for the postsmoother (when using Multigrid).
Presmoother		Handles settings for the presmoother (when using Multigrid).
Previous Solution		Previous parametric solution parameters.
Segregated		Handles parameters for a segregated solution approach.
Segregated Step		Handles settings for a segregated step (when using Segregated).
Sensitivity		Sensitivity parameters.

TABLE 5-2: SOLUTION ATTRIBUTE FEATURES

FEATURE	ICON	PURPOSE
SOR	SOR	Handles settings for an SOR-type linear solver or preconditioner.
SOR Gauge	SOR	Handles settings for an SOR Gauge-type linear solver or preconditioner.
SOR Line	SOR	Handles settings for an SOR Line linear solver or preconditioner.
SOR Vector	SOR	Handles settings for an SOR Vector-type linear solver or preconditioner.
Stationary Acceleration	🕒	Accelerates the solution process for nonlinear problems with a time-periodic stationary solution.
Stop Condition	🛑	Stop condition parameters.
Time Parametric	Pi= 213	Handles settings for parameter stepping. This attribute can be used together with the Time-Dependent Solver.
Vanka	V	Handles settings for a Vanka linear solver or preconditioner.

AVAILABLE SOLUTION UTILITY FEATURES

Table 5-3 lists the available Solution Utility Features:

TABLE 5-3: SOLUTION UTILITY FEATURES

FEATURE	ICON	PURPOSE
Adaptive Mesh Refinement		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	∂u/∂t	Compiles equations by specifying which study and study step to use and specifying representation of complex variables
Store Solution		Stores the solution at this point of the solver sequence also after the solver configuration has been computed.
State-Space		Provides a way of accessing state-space matrices for further work in Java.

Solution Operation Features

The following sections describe the solver operation features and their user interfaces in detail and includes these topics:

- AWE Solver
- Eigenvalue Solver
- Modal Solver
- Optimization Solver
- Stationary Solver
- Time-Dependent Solver
- Time Discrete Solver
- Time Explicit Solver
- Dependent Variables
- References for the Solution Operations Features

AWE Solver

Use the **AWE Solver** ( ) to perform fast-frequency parameter sweeps using asymptotic waveform evaluation (AWE).

Note

This an alternative way to perform parameter stepping to the one you get by using the **Stationary** solver node in conjunction with the **Parametric** subnode.

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.

Important

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.
- Use the **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.

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.
- Select 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** feature. 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 rather than 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 field **Parameter values** 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 is interrupted with an error if a too short interval is found. When the check box is selected 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**.

Eigenvalue Solver

Use the **Eigenvalue Solver** ( ) to find the solution to linear or linearized eigenvalue problems (also called eigenfrequency problems).

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).

Use the **Desired number of eigenvalues** field to specify the number of eigenvalues and eigenvectors that the solver should compute and store in the output. The number that can be specified in the **Relative tolerance** field controls the relative error in the computed eigenvalues.

The **Eigenvalue** transformation list you can select a transformation method for transforming the eigenvalues into another related quantity. You can always select **Eigenfrequency** or **None** (which keeps the original eigenvalues), but depending on the physics interfaces 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.

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** 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.
- Select any other available solution to use it as linearization point.

To store the used linearization point and the deviation from that linearization (instead of the total solution) in the output, select the **Store linearization point and deviation in output** check box.

If the eigenvalue itself appears nonlinearly, the solver reduces the problem to a quadratic approximation around an eigenvalue linearization point. Use the field **Eigenvalue linearization point** to specify such a scalar.

OUTPUT

Use the **Scaling of eigenvectors** list to specify the scaling method used to normalize the eigenvectors. Select:

- **RMS** to use root mean square (RMS) normalization.
- **Max** to use max 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 `mod1_u`).

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

ADVANCED

By default, the name of the eigenvalue is `lambda` when it appears in equations and expressions. Use the **Eigenvalue name** field to change this name.

The eigenvalue solver is an iterative algorithm. Use the **Maximum number of eigenvalue iterations** field to limit the number of iterations.

Use the **Dimension of Krylov space** field to control the algorithm's memory use. A 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.

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.

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.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Use the **Analysis type** list to select the basic analysis type. Select:

- **Frequency response** to perform parameter stepping using a reduced model.
- **Transient response** to perform time stepping using a reduced model.

Further options for Frequency Response

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 **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. Use the **Read File** button to read the specified file. The read values appear in the **Parameters values** field.

Important

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:

- **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.
- **Standard** to use the parameter values without modification.

Further Options for Transient Response

Use the **Times** field to enter a vector of times that define the time span for the simulation. Output from a simulation includes the times given in this field and the corresponding solutions.

TOLERANCES

Use the **Relative tolerance** field to enter a positive number. Depending on the selection in the **Analysis type** list in the **General** section, the tolerance has the following meanings:

- When **Analysis type** in the **General** section has been set to **Frequency response** 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 **Analysis type** in the **General** section has been set to **Transient response** the **Relative tolerance** is used by the solver in each time step to control the relative error.

EIGENPAIRS

- Use the **Solution** list to specify a solver sequence to be used when constructing the reduced model.
- Use the **Use** field to specify a data set containing the modes to be used in the reduced model. The selected data set must stem from the **Eigenvalue** operation feature.
- Use the **Modes** field to specify which of the eigenpairs present in the data set to include when constructing the reduced model. If the field is left empty, the solver uses all available eigenpairs.
- 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 either 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 left empty, no damping is applied by the solver.

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 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.
- Select any other available solution to use it as linearization point.

To store the used linearization point in the output, select the **Store linearization point and deviation in output** check box.

OUTPUT

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

ADVANCED

Use the **Load factor** field to enter a globally available scalar valued expression. 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).

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 either be one or equal to the number of eigenpairs in the reduced model as determined by the settings in the **Eigenpairs** section. If one number is entered, that value becomes the damping ratio for all eigenpairs.

Optimization Solver

Note

The **Optimization Solver** ( ) feature provides the settings necessary for solving PDE-constrained optimization problems. This feature requires a license for the Optimization Module. See the *Optimization Module User's Guide* for details.

Plugflow Solver

Note

The **Plugflow Solver** ( ) is a version of the Time Dependent solver specially designed for solving plug-flow reactor models set up in the Chemical Reaction Engineering Module's Reaction Engineering interface. This feature requires a license for the Chemical Reaction Engineering Module.

The **Plugflow Solver** ( ) is the default solver for Stationary Plug Flow study steps. It shares all settings with 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). For description of the available settings, see Time Dependent.

Stationary Solver

Use the **Stationary Solver** ( ) to find the solution to stationary problems (also called static or steady-state problems).

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

The number that can be specified in the **Relative tolerance** field is used for tolerance based termination of iterative solver processes and for error checking (if enabled) for direct linear system solvers.

Note

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 for attribute features.

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.
- **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 you select either **Linear** or **Linear perturbation** from the **Linearity** list, COMSOL Multiphysics assumes that the problem to be solved is a linearization about a solution. You can specify such a solution (a linearization point) using the **Prescribed by** list when **Linearity** has been set to **Linear** or **Linear perturbation.** 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.
- Select 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** feature. 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

Select the **Store reaction forces** check box if you want to compute and store reaction forces in the output.

LOG

This section, which is initially empty, contains a log from the time stepping. It is not available when the stationary solver is an attribute (subnode) to another solver.

Time-Dependent Solver

Use the **Time-Dependent Solver** ( ) to find the solution to time-dependent problems (also called dynamic or unsteady problems) using either of the implicit time stepping schemes BDF or the generalized-α method.

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 time span for the simulation.

Use the **Relative tolerance** field to enter a positive number. This number is used by the solver in each time step to control the relative error.

ABSOLUTE TOLERANCE

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.

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 either **Scaled** or **Unscaled** from the **Method** list additional fields appear. Use the **Tolerance** field 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 has been set to **BDF**, then it is necessary to specify absolute tolerances for time derivatives of the variables as well. 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.

Note

The **Method** setting (**Scaled** or **Unscaled**) that is selected for a variable applies also to its time derivative.

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

$$\left(\frac{1}{N}\sum_{i}\left(\frac{|E_i(U)|}{A_{us,\,i}+R|U_i|}\right)^2\right)^{1/2} < 1$$

where $A_{us,i}$ is the unscaled absolute tolerance for DOF i, R is the relative tolerance, and N is the number of degrees of freedom. The numbers $A_{us,i}$ are computed from the input value A_{k_i} for the corresponding dependent variable k_i. When using the time stepping method **BDF** and the **Scaled Method**, the step is accepted if

$$\left(\frac{1}{N}\sum_{i}\left(\frac{|E_i(Y)|}{A_{s,\,i}+R|Y_i|}\right)^2\right)^{1/2} < 1$$

where $A_{s,i}$ is the scaled absolute tolerance for DOF i, R is the relative tolerance, N is the number of degrees of freedom, and Y is the scaled solution vector. For dependent variables that are using the scaling method **Automatic**, the numbers $A_{k,i}$ are computed from the input values A_i according to the formula

$$A_{k,i} = A_{k_i}(\beta + \|Y\|_{2,i}) \qquad \beta = \begin{cases} (1-e^{-\alpha j})\|Y\|_{\infty,k_i} + e^{-\alpha j} & 0 < \|Y\|_{\infty,k_i} < 1 \\ 1 & \text{else} \end{cases}$$

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 max-norm of the dependent variable k_i respectively. For dependent variables that are using another scaling method, when the time stepping method is **Generalized alpha**, or when the method is **BDF** and the **Update scaled absolute tolerance** check box is cleared, then $A_{k,i} = A_{k_i}$. Note that this check box is placed under the **Time Stepping** section.

Caution

If the solution is smaller than the absolute tolerance, there is no accuracy at all. Because the displacements in most structural mechanics models are small, it is important to ensure that the absolute tolerance is smaller than the displacements.

Note

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 below for information about excluding algebraic equations from the error estimate.

TIME STEPPING

Select a time stepping method as well as various settings that apply to these methods.

Select a time stepping method with the **Method** list. 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 either a **Fully Coupled** or **Segregated** attribute feature is attached to a **Time** node, then the settings for the nonlinear systems solved by the time stepping methods are taken from that attribute feature.

Note

The time stepping method **Generalized alpha** requires either a **Fully Coupled** or **Segregated** attribute feature.

The time stepping method **BDF** can be used without a **Fully Coupled** or **Segregated** attribute feature. In such a situation the **BDF** method uses an internal and automatic nonlinear solver.

The **Steps taken by solver** list makes it possible to modify how the time stepping methods **BDF** and **Generalized alpha** select their time steps. Select:

- **Free** to let the time stepping method choose time steps freely. The times specified in the **Times** field in the **General** section is 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.
- (Only available for **Generalized alpha**) Select **Manual** to override the automatic choice of time step with a manual choice.

Note

This option also overrides the local error estimation made in each time step.

- **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.

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:

- **Initial step** (not available when **Steps taken by solver** is set to **Manual**). By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.
- **Maximum step** (not available when **Steps taken by solver** is set to **Manual**). By default the solver chooses a maximum time step automatically. Select the **Maximum step** check box for manual specification of a maximum time step.

- **Time step** (only available when **Steps taken by solver** is set to **Manual**). Enter a manual time step specification as a scalar, as a vector of times, or as an expression containing global expression variables in the **Time step** field.
- **Allow complex numbers** (only available when **Method** is set to **BDF**). 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** (only available when **Method** is set to **BDF**). This setting controls the maximum allowed degree of the interpolating polynomial of the BDF method.
- **Minimum BDF order** (only available when **Method** is set to **BDF**). This setting can be used to prevent the solver from decreasing the order of the BDF method below 2.

Note

This does not apply to the start-up phase of the simulation.

- **Update scaled absolute tolerance** (only available when **Method** is set to **BDF**). See **Absolute Tolerance**.
- **Time step increase delay** (only available when **Method** is set to **Generalized alpha**, but not when **Steps taken by solver** is set to **Manual**). Check this check box and enter a positive integer in the field if you want the solver to be more restrictive when increasing the time step. Entering 0 results in the same behavior as clearing the check box.
- **Amplification for high frequency** (only available when **Method** is set to **Generalized alpha**). By entering a number between 0 and 1 you can 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** (only available when **Method** is set to **Generalized alpha**). 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

Select the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depend on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time stepping method.

If applicable, use the **Probe** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

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.

Note

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 if you want to compute and store reaction forces in the output.

Select the **Store time derivatives** check box if you want 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 if you want the output solution to be stored on disk rather than in the computer's internal memory.

ADVANCED

Use the list **Singular mass matrix** to control whether the solver automatically determines if a system includes a differential-algebraic equation or not. Select:

- **Automatic** 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 list **Consistent initialization** 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.
- **Off** to indicate that the initial values already are consistent, which means that the solver does not modify them.
- (Only available when **Time method** has been set to **BDF** at the same time as the internal nonlinear solver of the **BDF** method is being used). Select **On** to use a consistent initialization routine that is preferable to **Backward Euler** for index-1 differential-algebraic equations.

Use the list **Error estimation** to control how to treat algebraic degrees of freedom of a differential-algebraic system when estimating the time discretization error. Select:

- **Include algebraic** to include the algebraic degrees of freedom in the error norm.
- **Exclude algebraic** to exclude the algebraic degrees of freedom from the error norm.

LOG

This section, which is initially empty, contains a log from the time stepping.

Time Discrete Solver

Use the **Time Discrete Solver** ( ) to find the solution to time-dependent problems (also called dynamic or unsteady problems) that have already been discretized in time using, for example, the `prev` operator or the `bdf` operator.

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 time span for the simulation.

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.

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 is used by the nonlinear 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.

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 either **Scaled** or **Unscaled** from the **Method** list additional fields appear. Use the **Tolerance** field to modify the absolute tolerance for the selected variable.

RESULTS WHILE SOLVING

Check the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time stepping method.

If applicable, use the **Probes** list to plot results from one or more probes. Select **All** to plot all probe results or **Manual** to manually choose which probe results to plot.

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.
- Select **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 rather than 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 classic Runge-Kutta explicit time stepping schemes.

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 time span for the simulation.

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.

Use the **Method** list to specify which family of explicit time stepping schemes to use. Currently, only the classic Runge-Kutta family is implemented.

Select the order of the time stepping scheme from the **Order** list.

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 are shown in the model tree as usual. For cheap, but approximate, inversion of the mass matrix use the **Lumped** option.

Note

This option can only be used together with a linear space discretization.

RESULTS WHILE SOLVING

Check the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time stepping method.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

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 rather than in the computer’s internal memory.

Dependent Variables

The **Dependent Variables** node ( ) 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** nodes ( ) under the **Dependent Variables** node. This feature 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. See Field for details about the settings available for a **Field** node. You can plot and evaluate the initial values for the dependent variables by right-clicking the **Dependent Variables** node and choose **Compute to Selected** or click the corresponding button ( ) on the **Settings** window's toolbar.

The **Settings** window for the **Dependent Variables** node contains the following sections:

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

INITIAL VALUES OF VALUES SOLVED FOR

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 interfaces 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.
- Select any available solution object to use it as initial value.

SCALING

Use the **Method** list to specify how to scale the variables solved for. Select:

- **Automatic** to get an automatically determined scaling.
- **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.
- **None** to get no scaling.

If you choose **Manual**, specify the scale by entering a scalar value in the **Scale** edit field.

The scaling method also can be specified per variable in the **Settings** window for the variable's **Field** node.

VALUES OF VARIABLES NOT SOLVED FOR

These settings are only available if there are dependent variables in the model that you do not solve for. 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 interfaces 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.
- Select any other available solution object to use it as initial value.

You select whether to solve for a variable or not by left-clicking a **Field** node and then check or clear the **Solve for this field** check box in the **Settings** window that appears.

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** (the default) to send the variables solved for, as specified by the **Field** node, as the solution data for results analysis.
- **Variables not solved for** to send the variables not solved for, as specified by the **Field** nodes, as the solution data for results analysis.

References for the Solution Operations Features

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. 99–13, 2005.

Solution Attribute Features

The following sections describe the solver attribute features and their user interfaces in detail and includes these topics:

- Adaptive Mesh Refinement
- Advanced
- Automatic Remeshing
- Coarse Solver
- Direct
- Field
- Fully Coupled
- Incomplete LU
- Iterative
- Jacobi
- Krylov Preconditioner
- Lower Limit
- Lumped Step
- Multigrid
- Parametric
- Postsmoother
- Presmoother
- Previous Solution
- Segregated
- Segregated Step
- Sensitivity
- SOR
- SOR Gauge
- SOR Line
- SOR Vector
- Stationary Acceleration

- Stop Condition
- Time Parametric
- Vanka

Adaptive Mesh Refinement

This is an attribute feature that adds adaptive mesh refinement. You can add an **Adaptive Mesh Refinement** node (▨) together with the **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver** operation features.

GENERAL

The software only performs adaptive mesh refinement in one geometry. Use the **Adaption in geometry** list to specify that geometry.

Use the **Maximum number of elements** edit 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.

Properties for the Stationary and Eigenvalue Adaptive Solver

Use the **Maximum number of refinements** edit field to specify the maximum number of mesh refinement iterations. The default value is 2 refinements.

Properties 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** edit 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**). 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: 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. Click the check box to specify a user-defined time interval length in the field (default value: 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 the check box to specify a user-defined minimal interval length in the field (default value: 0.1 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** (see below). 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 Estimates for the Stationary and Eigenvalue Solvers

Select **L2 norm** to use the L2 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. (Only available when using **Adaptive Mesh Refinement** together with the **Stationary** operation feature.)

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, one for each field variable. 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 L2 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 feature 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 when using **Adaptive Mesh Refinement** together with the **Eigenvalue** operation feature). For eigenvalue problems, the error estimate is a weighted sum of the error estimates for the various eigenmode. Use this edit 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 Estimates 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. A possible error indicator is the L_2 norm of the gradient of the dependent variables (for example, `sqrt(Tx^2+Ty^2)` for the temperature in a 2D heat transfer model).

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.

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 L2 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 field **Element fraction** 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 field **Element fraction** 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** node.

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 list **Consistent initialization** to control how the solver performs consistent initialization of differential-algebraic systems.

OUTPUT

This section contains information about which solution and mesh that contain the results from the adaptive mesh refinement.

The first time the solver runs the adaptive mesh refinement, the software creates a new solver sequence containing the adapted solution in an **Adapted Solution** utility feature. Subsequent adaptive solver calls can either generate new solver sequences or overwrite the last created one. Use the **Clear previous solution** check box to control whether to create new solver sequences with adapted solutions or not.

The **Clear previous mesh** check box works in the same way as the **Clear previous solution** check box and controls whether to create new mesh sequences with adapted meshes or not.

PLOT WHILE SOLVING

This section is available for the **Stationary Solver** and **Eigenvalue Solver** nodes.

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.

Advanced

The **Advanced** node ( ) is an attribute feature that handles advanced settings for operation features such as **Stationary Solver**, **Time-Dependent Solver**, and **Eigenvalue Solver**.

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.
- **Hermitian** to override the automatic symmetry detection and force the solver to assume that matrices are Hermitian.
- **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.

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. PDE problems 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 both symmetric and Hermitian matrices, so all you need to do to take advantage of the computational savings for a model with a symmetric matrix is to use a solver that utilizes the symmetry. The following linear system solvers and preconditioners do not take advantage of symmetric (or Hermitian) matrices:

- The Vanka preconditioner
- The Incomplete LU preconditioner
- The Algebraic multigrid solver/preconditioner

Caution

Selecting **Symmetric** for a problem that does not result in symmetric matrices leads to an incorrect solution.

Additional Solver Options

Use the **Row equilibration** check box to control whether row equilibration is performed or not. Even if this check box has been selected, the software might avoid to perform row equilibration in certain situations in order to preserve matrix symmetry.

Use the **Null-space function** list to select a method for the computation of matrices needed for constraint handling (see Elimination Constraint Handling below). 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.

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.

Use 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.

Use the **Stop when undefined mathematical operation is detected** check box to control how the solver handles undefined mathematical operations such as division by zero.

Elimination Constraint Handling

The constraint handling is for simplicity here 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 Multiphysics 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

$$D_c = \mathrm{Nullf}^T D\, \mathrm{Null}, \qquad E_c = \mathrm{Nullf}^T E\, \mathrm{Null}$$

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.

Reassembly of Loads

When the **Manual control of reassembly** check box is selected you can control reassembly of the residual vector with the **Constant load** list. Select:

- **On** to instruct the solver not to perform a reassembly process for the computation of the residual vector.
- **Off** to instruct the solver to perform a reassembly process for the computation of the residual vector.

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.

Even though **Constant load** has been set to **Off** some of the quantities that make up the residual might still be constant. Manual control of reassembly of these quantities can be controlled with the **Constant Jacobian**, **Constant damping or mass**, and **Constant mass** check boxes.

Select the **Jacobian constant** check box if the Jacobian matrix K is time independent. You can also choose this option if you want to use the same Jacobian throughout the time-dependent or nonlinear solver. This choice cuts down linear-system factorization/preconditioning time but causes more iterations because the Newton iteration is degraded into a fixed-point iteration.

Select the **Constant damping or mass** check box if the coefficients of the first-order time-derivative terms or the second-order time-derivative terms are time independent (often the case). In the discretized model, this means that the damping (sometimes called mass) matrix D or the mass matrix E is constant.

Reassembly of Constraints

When the **Manual control of reassembly** check box is selected you can control reassembly of the constraint residual with the **Constant constraint** list. Select:

- **On** to instruct the solver not to perform a reassembly process for the computation of the constraint residual vector.
- **Off** to instruct the solver to perform a reassembly process for the computation of the constraint residual vector.

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.

Even though **Constant load** has been set to **Off** the constraint Jacobian might still be constant. Manual control of reassembly of that quantity can be controlled with the **Constant constraint Jacobian** check box.

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.

Automatic Remeshing

This attribute feature adds automatic remeshing. You can add an **Automatic Remeshing** node ( ) together with the **Time-Dependent Solver** operation feature. It can be used together with the Moving Mesh interface to assure a satisfactory mesh quality throughout the simulation.

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.

MESH

Use the **Mesh quality expression** field to specify the expression that represents the mesh quality (such as `mod1.ale.minqual`). Then define the requested mesh quality (as a scalar number between 0 and 1) in the **Minimal mesh quality** field (default value: 0.2). When the mesh quality becomes smaller than requested the solver stops and remeshes at a previously stored solution time. The simulation then automatically continues using the new mesh from this solution time.

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 for manual specification of an initial step. Use the list **Consistent initialization** to control how the solver performs consistent initialization of differential-algebraic systems.

OUTPUT

This section contains information about which solution and meshes that contain the results from the automatic remeshing feature.

Coarse Solver

The **Coarse Solver** node ( ) is an auxiliary attribute feature used by the **Multigrid** attribute feature. This feature 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.

Direct

Attribute feature that handles settings for direct linear system solvers. You can use a **Direct** node ( ) together with operation features such as the **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver**. 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 feature. Although several attribute features for solving linear systems can be attached to an operation feature, only one can be active at any given time.

GENERAL

Use the **Solver** list to select a linear system solver. Select:

- **MUMPS** to use the MUMPS (MUltifrontal Massively Parallel sparse direct Solver).
- **PARDISO** to use the PARDISO solver.
- **SPOOLES** to use the SPOOLES (SParse Object Oriented Linear Equations Solver).

All the 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 non zeros in the L and U factors. Popular preordering algorithms include Minimum degree, Nested dissection, and Multisection. The solvers MUMPS and SPOOLES run distributed when running COMSOL 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.

Further Options for MUMPS

Use the **Memory allocation factor** field to tell the package how much memory is required. MUMPS estimates how much memory the unpivoted system requires. The **Memory allocation factor** is used to tell MUMPS how much more memory the pivoted system requires.

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Automatic** (automatically selected by the MUMPS package).
- **Approximate minimum degree**.
- **Approximate minimum fill**.
- **Quasi-dense approximate minimum degree**.
- **Nested dissection.**

Use the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not. Clear the check box to turn off the weight matching strategy.

Use the **Use Pivoting** list to control whether or not pivoting should be used.

When pivoting is used, use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold. 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.

Select the **Out-of-core** check box to store matrix factorizations as blocks on disk rather than in the computer's internal memory. When the check box is selected you can control the maximum amount of internal memory allowed for the blocks in the **In-core memory (MB)** field.

Further Options for PARDISO

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Minimum degree**.
- **Nested dissection**.

Use the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not. Clear the check box to turn off the weight matching strategy.

Use the **Bunch Kaufmann** check box to control whether to use Bunch-Kaufmann pivoting or not. Clear the check box to turn off Bunch-Kaufmann pivoting.

Use the **Pivoting perturbation** field to control the minimum size of pivots.

Select the **Out-of-core** check box to store matrix factorizations as blocks on disk rather than in the computer's internal memory. When the check box is selected you can control the maximum amount of internal memory allowed for the blocks in the **In-core memory (MB)** field.

Further Options for SPOOLES

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Best of ND and MS** (the best of nested dissection and multisection)
- **Minimum degree**
- **Multisection**
- **Nested dissection**

Use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold. 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 can be controlled by choosing the appropriate option from the **Check error estimate** list. **No** implies no error checking, whereas either **Automatic** or **Yes** does. 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. By default, error checking is enabled with the **Automatic** option meaning that the main solver has the responsibility for error management. No exceptions are thrown by the direct solver unless you choose the **Yes** option. If an error occurs in the main solver, warnings originating from the error checking in the direct solver appear.

Field

The **Field** node ( ) is an attribute feature that handles settings for field variables. Each field variable needs a separate **Field** node. This attribute is used together with the Dependent Variables operation feature.

GENERAL

Use the **Solve for this field** check box to control whether to solve for the field (variable) or not. This setting is only available if the **Dependent Variables** operation feature **Defined by study step** is **User defined**. If the variable is not solved for its values is determined by the settings in the **Variables Not Solved For** section of the corresponding **Dependent Variables** operation feature.

Use the **Store in output** check box to control whether to store the variable in any output solution or not. The value of a variable not solved for is controlled in the **Dependent Variables** node.

Note

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. 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 feature.
- **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.
- **None** to get no scaling.

If you choose **Manual**, specify the scale by entering a value in the **Scale** field.

Specifying a **Method** for a variable here overrides the **Method** selected in the **Scaling** section of the corresponding **Variables** operation feature unless **From parent** is selected.

Fully Coupled

Attribute feature that handles parameters for a fully coupled solution approach. The **Fully Coupled** node ( ) uses a damped version of Newton's method, and you can use it together 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 features can be attached to an operation feature, only one can be active at any given time.

GENERAL

Use the **Linear solver** list to select a solver for linear systems that appear in the corresponding solver sequence. The available solvers are attribute features of the types **Direct** and **Iterative**.

DAMPING AND TERMINATION

Use the **Damping method** list to control which damping factor to use in the damped Newton iterations:

- Select **Automatic** to let the solver automatically determine a damping factor in each iteration of Newton's method.
- Select **Constant** to manually specify a constant damping factor that is used in all iterations of Newton's method.
- The option **Automatic highly nonlinear** is similar to **Automatic** with the difference that this option can make the solver more careful when solving highly nonlinear problems. Try this option if the solver does not converge with **Automatic**.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Damping method** list are:

- **Initial damping factor** (not available when **Damping method** is set to **Constant**). Use this field to specify a damping factor for the first Newton iteration.
- **Minimum damping factor** (not available when **Damping method** is set to **Constant**). Use this field to specify the smallest allowed damping factor.
- **Restriction for step-size update** (not available when **Damping method** is set to **Constant**). Use this field 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.
- **Use recovery damping factor** (not available when **Damping method** is set to **Constant**). If a damping factor smaller than the **Minimum damping factor** is required, the nonlinear solver terminates when **Off** is selected in this list. When **On** is selected, the nonlinear solver takes a Newton step using the constant damping factor that you enter in the **Recovery damping factor** field. The default setting **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 the subsequent parameter values.
- **Damping factor** (only available when **Damping method** is set to **Constant**). Use this field to specify a constant damping factor for Newton's method.
- **Jacobian update** (only available when **Damping method** is set to **Constant**). The available options in this list are **Minimal**, **On every iteration**, and **On first iteration**. The option **On every iteration** computes a new Jacobian for all iterations of Newton's method. The two options **Minimal** and **On first iteration** are identical when used together with a **Stationary** operation feature. They both compute a new Jacobian for

the first iteration of Newton's method only. The option **On first iteration** computes a new Jacobian for each new nonlinear system of equations when used with a **Time-Dependent Solver** while **Minimal** reuses the Jacobian for several nonlinear systems whenever deemed possible.

Select the **Limit on nonlinear convergence rate** check box (only available when used together with a **Time-Dependent Solver** and when **Damping method** is set to **Constant**) 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.

Use the **Termination technique** list 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.
- Select **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.
- (Only available when **Damping method** has been set to **Constant**). Select **Iterations** to terminate the Newton iterations after a fixed number of iterations.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are:

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field 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.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field 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** nodes.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). Use this field to specify a fixed number of iterations to perform.

RESULTS WHILE SOLVING

Select the **Results While Solving** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

Incomplete LU

Attribute feature that handles parameters for linear system solvers/preconditioners that use incomplete LU factorization. You can use the **Incomplete LU** node ( ) together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features. Right-click any of these nodes to add the feature.

GENERAL

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.

Drop Using

For **Incomplete LU**, select an option from the **Drop using** list to specify 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. A larger tolerance neglects more elements.

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. 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.

Number of Iterations and Relaxation Factor

For **Incomplete LU**, the incomplete LU factorization can be applied iteratively. Use the **Number of iterations** field to specify how many iterations to perform. The iterative process uses a relaxation factor similar to the one used be, for example, the **SOR** attribute feature. Specify such a factor in the **Relaxation factor** field.

Preordering Algorithm

For **SPOOLES**, use the **Preordering algorithm** list to select one of the following preorderings:

- **Best of ND and MS** (the best of nested dissection and multisection)
- **Minimum degree**
- **Multisection**
- **Nested dissection**

Iterative

Attribute feature that handles settings for iterative linear system solvers. You can use the **Iterative** node (⧅) together with operation features such as the **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver**. An alternative to the iterative linear system solvers is given by direct linear system solvers, which are handled via the **Direct** attribute feature. Although several attribute features for solving linear systems can be attached to an operation feature, 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 feature that this attribute belongs to.

GENERAL

Use the **Solver** list to select a linear system solver:

- Select **GMRES** to use the restarted GMRES (Generalized Minimum RESidual) iterative method.
- Select **FGMRES** to use the restarted FGMRES (Flexible Generalized Minimum RESidual) iterative method.
- Select **BiCGStab** to use the BiCGStab (BiConjugate Gradient Stabilized) iterative method.
- Select **Conjugate gradients** to use the conjugate gradients iterative method.
- Select **Use preconditioner** to use the currently active preconditioner attached to this **Iterative** attribute feature 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** attribute feature.

It is possible to roughly order the linear system solvers according to their memory usage and computational time per iteration (with least memory and time first):

1. Conjugate gradients
2. BiCGStab
3. GMRES
4. FGMRES

The solvers that require less memory and computational time per iteration typically are less robust and not applicable to all problem types.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Solver** list are:

- **Number of iterations before restart** (only available when **Solver** is set to either **GMRES** or **FGMRES**). Use this field to specify how many iterations the solver should take between each restart. A larger number increases robustness but also memory use and computational time.
- **Preconditioning** (not available when **Solver** is set to **FGMRES**). Use this 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.

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 feature 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. A larger value of ρ increases the number of iterations, but decrease the chance that the iteration is terminated too early.

Use the **Maximum number of iterations** field to enter a maximum number of iterations that the iterative solver is allowed to take. When this number of iterations has been performed without reaching the specified tolerance the solver is automatically stopped 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 either **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.

Jacobi

Attribute feature that handles settings for the **Jacobi** ( ) (or diagonal scaling) method. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features.

GENERAL

Settings When Not Used With Coarse Solver

Enter the **Number of iterations** to perform when this attribute feature is being 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 feature). The solver then iterates until the relative tolerance specified by the corresponding operation feature is fulfilled rather than performs a fixed number of iterations.

Enter a **Relaxation factor** to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. The default is 1.

Settings When Used with a Coarse Solver

If this feature 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:

- The **Number of iterations** to perform. The default is 10.
- The **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:

- The **Relative tolerance** to specify the termination tolerance. The default is 0.1.
- The **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.
- The **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

Attribute feature that handles settings for Krylov-type methods. You can use the **Krylov Preconditioner** node ( ) together with the **Iterative**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features. This feature contains settings similar to the ones for the Iterative attribute feature. The difference is that this feature is an auxiliary attribute feature whereas **Iterative** handles settings for a main iterative solver. This feature can also be used with another **Krylov Preconditioner** attribute feature.

GENERAL

Select a linear system **Solver**. Select:

- **GMRES** to use the restarted Generalized Minimum RESidual iterative method.
- **FGMRES** to use the restarted Flexible Generalized Minimum RESidual iterative method.

- **BiCGStab** to use the BiConjugate Gradient Stabilized iterative method.
- **Conjugate gradients** to use this iterative method.

Note

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.

Number of Iterations before Restart

When **GMRES** or **FGMRES** is selected as the **Solver**, 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.

Preconditioning

When **GMRES**, **BiCGStab**, or **Conjugate gradients** are selected as the **Solver**, select an option from the **Preconditioning** list to specify whether to precondition the linear system matrix from either the **Left** or from the **Right**. Normally, this setting does not significantly influence the convergence behavior of the selected solver.

Termination Technique

For all **Solver** types, and to determine how to terminate the solver, select an option from the **Termination technique**— **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the solver is used, or **Use tolerance**, to terminate the solver when a tolerance is fulfilled.

- If **Fixed number of iterations** is selected, enter a **Number of iterations** to perform.
- If **Use tolerance** is selected, enter a **Relative tolerance** to specify the termination tolerance.

Number of Iterations

For all **Solver** types, enter a **Number of iterations**. The default is 10.

Maximum Number of Iterations

When **Use tolerance** is selected as the **Termination technique**, enter a **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.

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 either **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. These restrictions are checked after each substep of a segregated approach. The restriction is performed without any regards to the equations. You can use this attribute together with the **Segregated** attribute feature.

LOWER LIMIT

By default, no variables have active lower limits. To activate or inactivate a lower limit for a variable, select the variable in the **Variables** window. Then, use the **Method** list to select either **Active** or **Inactive** for that variable. When **Active** has been selected in the **Method** list you can use the **Lower limit (field variable)** field to specify a scalar lower limit.

Lumped Step

The **Lumped Step** node ( ) is available when using the segregated attribute feature. 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, or mass, 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, the user has to take care to ensure that the resulting approximate Jacobian is invertible.

Multigrid

The **Multigrid** solver ( ) is used to set up a geometric multigrid (GMG) solver or an algebraic multigrid (AMG) solver. Use can use the **Multigrid** solver together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features. Right-click any of these nodes to add the feature.

GENERAL

Select a **Solver**—**Geometric multigrid** or **Algebraic multigrid**. For both options, enter the **Number of iterations** and select a **Multigrid cycle**—**V-cycle** (the default), **W-cycle**, or **F-cycle**.

Note

The Multigrid cycle settings are the same as for the GMG and AMG solvers. The other settings are explained next.

Geometric Multigrid

If **Geometric multigrid** is selected, also select one of the following options from the **Hierarchy generation method** list to select how to generate the meshes for the multigrid levels:

- **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.
- **Lower element order first (any)**. 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**. 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**. Selecting multigrid level from the existing ones.

- Enter the **Number of multigrid levels** to generate (the default is 1).
- In the **Use hierarchy in geometries** box, select the geometry to apply multigrid to.

- 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 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 multigrid hierarchy generation method. The options are:

- **Split longest side**. 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.

Algebraic Multigrid

If **Algebraic multigrid** is selected, also enter the **Number of multigrid levels** to select how many multigrid levels to generate. The default is 5.

Enter a **Maximum number of DOFs at coarsest level**. The default is 5000.

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. The range goes from 1 to 10 where 10 gives the best quality. The default is 3.

Parametric

The **Parametric** node (Pi= 213) is an attribute feature that handles settings for parameter stepping. This attribute can be used together with a **Stationary Solver** feature.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Use the **Parameter names** field to specify one or several parameter names. If more than one parameter name is used enter them as a space-separated list.

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. If more than one **Parameter name** has been specified the list of parameter values is interpreted as follows: Assume that `p1` and `p2` are the parameter names and that `1 2 3 4` is the parameter list. The solver then first solves using `p1` equal to `1` and `p2` equal to `2`. Thereafter, it solves using `p1` equal to `3` and `p2` equal to `4`.

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. Reading parameter values from file overwrites any values already present in that field. The format of the text files must be such that the parameter values for each parameter appear columnwise with either a space or tab separating the values on each row.

Use the **Predictor** list to control how the initial value for the next parameter value is determined. Select:

- **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.

Note

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).

TUNING

By default the solver selects the parameter steps automatically based on the values entered in the **Parameter values** field in the **General** section. To be able to modify the default behavior select the **Tuning of step size** check box. Then the following options become available. Use the:

- **Initial step size** field to enter a positive number that determines the magnitude of the first parameter step.

- **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.
- **Minimum step size** field to specify a safeguard against too small parameter steps.

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 dictated 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 rather than in the computer's internal memory.

RESULTS WHILE SOLVING

Check the **Results While Solving** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select **All** to plot all probe results or **Manual** to manually choose which probe results to plot.

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.

Postsmoother

The **Postsmoother** node (🔄) is an auxiliary attribute feature used by the **Multigrid** attribute feature. This attribute does not have any settings. Instead, its purpose is to administrate postsmoothers for a multigrid solver. To add a postsmoother, right-click the **Postsmoother** node.

Presmoother

The **Presmoother** node ( ) is an auxiliary attribute feature used by the **Multigrid** attribute feature. This attribute does not have any settings. Instead, its purpose is to administrate presmoothers for a multigrid solver. To add a presmoother, right-click the **Presmoother** node.

Previous Solution

The **Previous Solution** node ( ) is an optional attribute feature of the **Parametric** attribute feature.

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 features of the types **Direct** and **Iterative**.

Segregated

The **Segregated** node ( ) is an attribute feature 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 feature. Although several **Fully Coupled** and **Segregated** attribute features can be attached to an operation feature, 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 they are shown in the model tree.

GENERAL

Use the **Termination technique** list to control how the segregated iterations are terminated. Select:

- **Tolerance** 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 that 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.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are:

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field to limit the number of segregated iterations. When the maximum number of iterations have been performed the segregated method is terminated even if the tolerance is not fulfilled.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field 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.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). Use this field to specify a fixed number of iterations to perform.

Select the **Limit on nonlinear convergence rate** check box (only available when used together with a **Time-Dependent Solver** node) 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.

RESULTS WHILE SOLVING

Check the **Results While Solving** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select **All** to plot all probe results.

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 feature.

GENERAL

Use the **Variables** list to specify variables to be solved for in this segregated step.

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 features of the types **Direct** and **Iterative**.

DAMPING AND TERMINATION

Use the **Damping method** list to control which damping factor to use in the damped Newton iterations. Select:

- **Automatic** to let the solver automatically determine a damping factor in each iteration of Newton's method.
- **Constant** to manually specify a constant damping factor that is used in all iterations of Newton's method.
- **Automatic highly nonlinear** for highly nonlinear problems. This option is similar to **Automatic** but can make the solver more careful when solving highly nonlinear problems. Try this option if the solver does not converge with **Automatic**.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Damping method** list are:

- **Damping factor** (only available when **Damping method** is set to **Constant**). Use this field to specify a constant damping factor for Newton's method.
- **Jacobian update** (only available when **Damping method** is set to **Constant**). The available options in this list are **Minimal**, **On every iteration**, and **On first iteration**. The option **On every iteration** computes a new Jacobian for all iterations of Newton's method. The two options **Minimal** and **On every iteration** are identical when used together with a **Stationary** operation feature. They both compute a new Jacobian for the first iteration of Newton's method only. The **On first iteration** option computes a new Jacobian for each new nonlinear system of equations when used with a **Time-Dependent Solver** node while **Minimal** reuses the Jacobian for several nonlinear systems whenever deemed possible.
- **Initial damping factor** (not available when **Damping method** is set to **Constant**). Use this field to specify a damping factor for the first Newton iteration.
- **Minimum damping factor** (not available when **Damping method** is set to **Constant**). Use this field to specify the smallest allowed damping factor.
- **Restriction for step-size update** (not available when **Damping method** is set to **Constant**). Use this field 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.

- **Use recovery damping factor** (not available when **Damping method** is set to **Constant**). If a damping factor smaller than the **Minimum damping factor** is required, the nonlinear solver terminates when **Off** is selected in this list. When **On** is selected, the nonlinear solver takes a Newton step using the constant damping factor that you enter in the **Recovery damping factor** field. The default setting **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 the subsequent parameter values.

Use the **Termination technique** list 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.
- **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.
- (Only available when **Damping method** has been set to **Constant**). Select **Iterations** to terminate the Newton iterations after a fixed number of iterations.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are:

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field 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.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field 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** nodes.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). Use this field to specify a fixed number of iterations to perform.

Sensitivity

A **Sensitivity** solver node ( ) solves a sensitivity analysis problem set up in a Sensitivity interface. The **Sensitivity** form contains a single section:

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. The adjoint method requires only one additional linear system solution.
- **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. The forward method requires one extra linear system solution for each sensitivity variable.

SOR

The **SOR** node ( ) handles settings for the SOR (successive over-relaxation) iterative method. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features. Right-click any of these nodes to add the SOR feature.

GENERAL

Settings When Used With Any Attribute Feature

Use the **Solver** list to specify which variant of the SOR algorithm to use. Select:

- **SSOR** to use the symmetric SOR algorithm which in each iteration performs one **SOR** sweep followed by one **SORU** sweep. This has the effect that the preconditioning matrix is symmetric whenever the system matrix is symmetric, which is important, for example, when preconditioning the conjugate gradients iterative method.
- **SOR** to use the forward SOR algorithm.
- **SORU** to use the backward SOR algorithm.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2.

Select the **Blocked version** check box to use a version of the SOR method which is optimized for parallel computations.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute feature is being used as a preconditioner or smoother. This setting is not considered when the attribute is being used as a linear system solver (with the **Use preconditioner** option in the **Solver** list of the **Iterative** attribute feature). The solver then iterates until the relative tolerance specified by the corresponding operation feature is fulfilled rather than performs a fixed number of iterations.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** 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.

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**). Use this field to specify the fixed number of iterations to perform.
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance**). Use this field to specify the termination tolerance.
- **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. 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.

SOR Gauge

The **SOR Gauge** node (SOR) handles settings for the SOR gauge iterative method. This is a method of SOR-type with added functionality useful for, for example, 3D magnetostatics. In short, the added functionality consists of divergence cleaning for degrees of freedom discretized with vector elements. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features.

MAIN

Settings When Used With Any Attribute Feature

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** 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.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2.

Select the **Blocked version** check box to use a version of the SOR Gauge method which is optimized for parallel computations.

Use the **Variables** list to specify variables to include in the divergence cleaning phase of an SOR Gauge iteration.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute feature is being used as a preconditioner or smoother.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** 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.

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**). Use this field to specify the fixed number of iterations to perform.
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance**). Use this field to specify the termination tolerance.
- **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. 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.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of divergence cleaning iterations to perform for each main iteration.

SOR Line

The **SOR Line** node (SOR) 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 attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features.

MAIN

Settings When Used With Any Attribute Feature

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2.

Use the **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, 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.

Select the **Blocked version** check box to use a version of the SOR method which is optimized for parallel computations.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute feature is being used as a preconditioner or smoother.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** 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.

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**). Use this field to specify the fixed number of iterations to perform.
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance**). Use this field to specify the termination tolerance.
- **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. 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.

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.

Use the **Relaxation factor** field to specify a scalar relaxation factor for the iterations specified in the **Number of secondary iterations** field. The allowed values of this factor are between 0 and 2.

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 curl-curl operator and vector elements. In short, the added functionality consists of performing SOR iterations on an auxiliary linear system in addition to the ordinary SOR iterations. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features.

MAIN

Settings When Used With Any Attribute Feature

Use the **Solver** list to specify which variant of the SOR Vector algorithm to use. Select:

- **SSOR vector** to perform one ordinary **SOR** iteration followed by a number of **SSOR** iterations on an auxiliary system and 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.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2.

Select the **Blocked version** check box to use a version of the SOR Vector method which 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 When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute feature is being used as a preconditioner or smoother.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** 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.

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**). Use this field to specify the fixed number of iterations to perform.
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance**). Use this field to specify the termination tolerance.
- **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. 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.

SECONDARY

Use the **Number of secondary iterations** field to specify the number iterations to perform the auxiliary system for each main iteration.

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. Furthermore, stationary acceleration can be used to speed up the solution process for some types of plasma models

Note

The **Stationary Acceleration** feature is not used per 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 (🛑) stops the solver when a specified condition is fulfilled. It is an optional attribute feature of the **Parametric** attribute feature and the **Time-Dependent Solver** operation feature.

STOP CONDITION

Use the **Stop expression** field to specify a scalar expression for the condition that defines when the solver should stop. The solver evaluates the expression after each time step or parameter step. The stepping stops if the real part of the expression becomes negative. The solver does not return the corresponding solution for which the expression is negative. See Adding a Stop Condition in the *COMSOL Multiphysics User's Guide* for an example. For a stop condition that includes more than one condition, you can use the `if` operator. For example, `if(A>0 && B<1E-10,-1,1)` stops when both conditions are fulfilled. The `if` operator then returns −1, and the solver stops. An example of a stop condition is `mod1.ale.relVolMin-0.01`, which makes the solver stop when the relative element volume in a Moving Mesh (ALE) interface in Model 1 becomes less than 0.01.

Time Parametric

The **Time Parametric** node ( ) is an attribute feature that handles settings for parameter stepping. This attribute can be used together with a **Time-Dependent Solver** or other time-dependent solver feature. The functionality when used as a subnode to a **Time-Dependent Solver** is similar to the **Parametric** as a subnode to a **Stationary Solver** with the following important differences:

- Continuation is not supported. The initial data is the same for all parameters.
- The output solution is of the parametric type and not the transient type, which make the data difficult to use for postprocessing.
- This node is never used automatically, but its time-parametric core functionality is used by the least-squares optimization solver in the Optimization Module.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Use the **Parameter names** field to specify one or several parameter names. If more than one parameter name is used enter them as a space-separated list.

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. If more than one **Parameter name** has been specified the list of parameter values is interpreted as follows: Assume that `p1` and `p2` are the parameter names and that `1 2 3 4` is the parameter list. The solver then first solves using `p1` equal to `1` and `p2` equal to `2`. Thereafter, it solves using `p1` equal to `3` and `p2` equal to `4`.

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. Reading parameter values from file overwrites any values already present in that field. The format of the text files must be such that the parameter values for each parameter appear columnwise with either a space or tab separating the values on each row.

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.

Vanka

The **Vanka** node ( ) handles settings for the Vanka iterative method. Formally, the method applies to saddle-point problems. The corresponding linear system matrix is indefinite and its diagonal often contains zeros. A typical example of a problem of this type 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 can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute features.

MAIN

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute feature is being used as a preconditioner or smoother.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** 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.

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**). Use this field to specify the fixed number of iterations to perform.
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance**). Use this field to specify the termination tolerance.
- **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. 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.

Settings When Used With Any Attribute Feature

Use the **Variables** list to specify variables to include in a Vanka block approach.

Use the **Block solver** list to specify how to solver the Vanka block linear systems. Select:

- **Direct** to use a direct solver.
- **GMRES** to use the iterative method GMRES.

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.
- **Number of iterations before restart** field to specify how many iterations the solver should take between each restart.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2.

Select the **Blocked version** check box to use a version of the Vanka method which 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. The allowed values of this factor are between 0 and 2.

Solution Utility Features

The following sections describe the solver utility features and their user interfaces in detail and includes these topics:

- Adaptive Mesh Refinement
- Assemble
- Compile Equations
- Store Solution
- State Space

Adaptive Mesh Refinement

The **Adaptive Mesh Refinement** ( ) utility feature is a simple container for a solution obtained using the **Adaptive Mesh Refinement** attribute feature. This utility node is added automatically (to an otherwise empty solver sequence) by the corresponding **Adaptive Mesh Refinement** attribute node. It is not possible to add this node manually, and it does not have any settings.

See Also

- Adaptive Mesh Refinement (attribute feature)

Assemble

The **Assemble** node ( ) can be used to access the raw data of any assembled matrix or vector from Java. Right-click the solver node and select **Other>Assemble**. Select the appropriate check boxes for the matrices and vectors you want to inspect or manipulate and save the model as a Model Java-file. 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.

Note

See the *COMSOL Java API Reference Guide* for details about the solver API for **Assemble**.

Compile Equations

In the **Compile Equations** ( ) node's **Settings** window you specify which study and study step to use when computing the current solver sequence and compiling the equations to solve. The node displays the name of the selected study step: **Compile Equations: Stationary**, for example.

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.

SPLIT COMPLEX VARIABLES

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 nonanalytical 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 interface nodes.

Note

The split representation enables a correct evaluation of Jacobians for the following operators: `real`, `imag`, `conj`, `abs`, and `realdot`.

Store Solution

This is a utility feature whose purpose is to make it possible to access intermediate solution results. By default, the software only stores the solution at the last computed feature of a solver sequence. Add a **Store Solution** node ( ) anywhere in a solver sequence to make the software store the solution at that point of the sequence in addition to the solution at the last computed feature. You can, for example, use the solution from a **Store Solution** node when analyzing the results and as initial values (via a study-type feature) for other computations.

State Space

The **State Space** node ( ) can be used to access the raw data of the state-space form of a dynamic PDE. To do so, right-click the solver node and select **Other>State Space**. Select the appropriate check boxes for the matrices and vectors you want to compute and save the model as a Model Java-file. 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.

Note

See the *COMSOL Java API Reference Guide* for details about the solver API for **State Space**.

Job Configurations

The **Job Configurations** node ( ) is visible when you click the **Show** button ( ) and select **Advanced Study Options**. The node automatically displays if the **Job Configuration** node has content.

See Also

- Show Advanced Study Options in the *COMSOL Multiphysics User's Guide*

There are three categories of job sequences, each of which corresponds to a model feature:

- Parametric
- Batch
- Cluster Computing

The first two categories share the common set of subfeatures listed in Table 5-4. In addition, a Batch node has a default Batch Data subnode that in turn stores External Process nodes, which contain information about batch jobs. Cluster Computing nodes support no subnodes and must point to a Batch node.

TABLE 5-4: PARAMETRIC AND BATCH SUBFEATURES.

FEATURE	ICON	PURPOSE
Job Configuration		Runs a Parametric, Batch, or Cluster Computing job feature
Solver		Runs a solver sequence
External Class		Runs the main method of an external Java class
Geometry Sequence		Runs a geometry sequence
Meshing Sequence		Runs a mesh sequence
Save Model to File		Saves the current state of the model on file
Plot Group		Runs a Plot Group feature

TABLE 5-4: PARAMETRIC AND BATCH SUBFEATURES.

FEATURE	ICON	PURPOSE
Derived Value	8.85 e-12	Runs a Derived Values feature
Export to File	⇨	Runs an Export feature and saves it on file

The **Job Configurations** node's context menu has these options:

- **Show Default Solver** ( )—Shows the default job sequence node (if any) that corresponds to the study step nodes in the study.
- **Parametric** ( )—Adds a **Parametric** node.
- **Batch** ( )—Adds a **Batch** node and an associated **Batch Data** subnode.
- **Cluster Computing** ( )—Adds a **Cluster Computing** node.
- **Delete Solvers** ( )—Deletes all jobs under the **Job Configurations** node.

Parametric

Parametric ( ) is one of the main features. It has the ability to loop over a given set of parameters. For each set of parameters it runs the sequence defined by its subfeatures. You can combine the sequence with other **Batch**, **Parametric**, or **Cluster Computing** sequences in a hierarchical way by adding a job sequence that points to another feature. You can, for instance, create a **Parametric** node that does a LiveLink update and then runs a **Cluster Computing** feature 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** feature 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 etc.) and variation over the parametric sweep. For independent variation of parameters the accumulated table can be used with the **Format: Filled** to manipulate the table data into a matrix format that can be used for so called 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 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.

Job Configuration

You can add a **Job Configuration** node ( ) to run another job sequence. You select the sequence to run form the **Configuration** list. Recursive calls are detected and cause errors.

Solver

The **Solver** feature ( ) runs a solver sequence. Select the sequence to run from the **Sequence** list. The default behavior is to run all solver sequences. Store a copy of the solution once it has run by selecting the **Keep all solutions** box. The name of the copy is generated from the current sequence name and parameter values. The default is to clear previous solutions. It can be disabled by clearing the **Clear previous** check box.

External Class

The **External Class** node ( ) runs the main method of an external compiled Java class. You add it by right-clicking a job sequence node and selecting it from the **Other** submenu. 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**.

Geometry Sequence

The **Geometry Sequence** node ( ) runs a geometry sequence. You add it by right-clicking a job sequence feature node and selecting it from the **Other** submenu. You select the sequence to run from the **Run** list. The default is to run all geometry sequences.

Meshing Sequence

The **Meshing Sequence** node ( ) runs a mesh sequence. You add it by right-clicking a job sequence feature node and selecting it from the **Other** submenu. You select the sequence to run from the **Run** list. The default is to run all meshing sequences.

Save Model to File

The **Save Model to File** node (💾) stores the model in the state it is at that point in the sequence. You select where to store the model by choosing a **Filename**. If you select the **Clear previous** check box any previous models with the same name are overwritten. If you are running a parametric sweep the model gets a unique name based on the current parameter values. The names of the saved models are stored in the **Filename** table. You can open a saved model in a new instance of COMSOL Multiphysics using the **Open** button.

Plot Group

The **Plot Group** node ( ) runs a sequence of plot groups, creating a plot in the Graphics window. You add it by right-clicking a job sequence feature node and selecting it from the **Results** submenu. You select the sequence to run in the **Run** field. The default behavior is to run all plot group nodes. You can use an **Export to File** node to store the resulting plot in a file.

Derived Value

The **Derived Value** node (8.85 e-12) runs a **Derived Values** node defined in the **Results** branch of the model tree. You add it by right-clicking a job configurations feature node and selecting it from the **Results** submenu. The computed value is stored in a **Result** table under the **Derived Value** feature. You select the node to run in the **Run** field. The default behavior is to run all **Derived Values** nodes. The **Parameters** column in the **Result** table contains the name of the **Derived Values** node that computed the numerical value. The **Table** setting decides which **Table** under **Results** to store the computed values in.

Export to File

The **Export to File** node ( ) runs an Export node defined in the **Export** branch of the model tree. You add it by right-clicking a job configurations feature node (**Batch**, for example) and selecting it from the **Results** submenu. 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. Use the **Run** setting to select the Export node to run. The default behavior is to run all Export nodes.

In the **File** section you set if COMSOL should overwrite files with the same name or if an error should occur. 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.

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.

Batch

The **Batch** node ( ) is the main feature 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. As in the **Parametric** 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. The **Batch** job has a special Batch Data subnode, which collects External Process subnodes containing job status information.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

You set the number of cores 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 do not exceed the number of cores available on the computer. Otherwise you experience performance degradation. You can set a **Start time** if you wish the batch process to start a later time. Select the hour to start the run in **Start time**.

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. Specify the **Directory** to store the model. If the batch job has another path to the directory you select the **Specify batch directory** check box and enter the path to the batch **Directory** or **Browse** for the path. If COMSOL is installed in a different directory where the batch job runs you enable the **Specify COMSOL install directory** 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, for instance. Click the **Save As Default** button on the toolbar to save the current directory setting as the default directory for batch files.

CLIENT SETTINGS

The **Batch** feature 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** and **Port number** of the server for the batch job to connect to.

Batch Data

The **Batch Data** node ( ) contains information about the batch processes that have been started by the **Batch** node. Each process is represented by a subnode called **External Process**. Click the **Show All Progress** button ( ) on the **Settings** window's toolbar to display the progress of all the **External Process**. The GUI enters a progress mode in order to follow the progress of the external processes. You can at any time stop the progress mode by clicking the **Stop Logging** button.

External Process

The **External Process** nodes contain information about the batch processes that have been started by the **Batch** node. Each **External Process** node is associated with a started batch job. The **Start command** field contains the command that was used to start the batch job. The **Filename** contains the file name of the model that is used in the batch job. When the batch job has finished you can open it using the **Open** button. You **Update log**, **Show progress** of the external job, **Cancel** the process, **Stop** the process, **Clear**

status of the external process, and **Rerun** the job, by choosing the **Operation** and **OK** buttons. The status of the operation is viewed in the status. The log is updated when you choose **Update log**. When you **Show progress** the COMSOL Desktop enters a progress mode in order to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Stop Logging** button.

Cluster Computing

The **Cluster Computing** node ( ) is useful when you want to submit a batch job to a job scheduler or you 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 **Options>Preferences>Cluster computing** section.

See Also

- Editing Preferences Settings in the *COMSOL Multiphysics User's Guide*

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 (see below).

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

You choose which type of cluster job you want to do from the **Cluster type** list:

- The **General** type can be configured to run on many types of clusters.

- Select **WCCS 2003** to use the Windows Compute Cluster Server 2003 job scheduler to submit the batch job.
- Select **HPCS 2008** to use the Windows HPC Server 2008 job scheduler to submit the batch job.

The **Not distributed** choice is useful when you want to submit a batch job to a job scheduler without running a distributed job. If you are using the **General** type and you have started a MP daemon on the computer select the **MPD is running** check box. Use the **Host file** field to specify the host file used for the job. If your cluster is Linux and it requires ssh (secure shell) or rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol. You set the number of computational nodes to use in the **Number of nodes** field. The number of processes on each node is set in the **Number of processes on host** field; usually this is 1.

In the **WCCS 2003** and **HPCS 2008** cases you can select the **Exclusive nodes** check box to use exclusive nodes. In addition, under **Advanced** you can specify the name of the scheduler host in the **Scheduler** field. You can also set names of **Requested nodes**, the **Runtime (minutes)** before the job is cancelled, the **User** name, and the **Priority** (highest, above normal, normal, below normal, or lowest) of the scheduled job. In addition, for **WHPC2008**, you can also set the **Node granularity** (node, socket, or core), the minimum required **Cores per node**, and the minimum required **Memory per node (MB)**.

Using a Job Sequence to Store Parametric Results on File

As an example of the use of a job sequence, add a **Parametric** node under **Job Configurations** 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 output 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 sequence 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 the parametric sweep node you created in **Parameter names** and set the **Parameter values** you want to sweep over. In some cases COMSOL will choose to use the more efficient parametric solver when sweeping. Set the **Use parametric solver** under **Extension** 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**. The models created during the simulation can then be found in a **Save model to File** node under **Job Configurations**. Note that the default settings are configurable in the Preferences dialog box.

3 If you want to create output files for further analysis, right-click the main study node and select **Show Default Solver**. Select the parametric node under **Job Configurations** that corresponds 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. Right-click and select **Results**. You can add a **Plot Group** node if you want to run a **Plot Group** from **Results** for each sweep. In most cases you should use the **Plot** settings under **Results While Solving** instead. Add a **Derived Value** node if you want to run a **Derived Values** node under **Results**. 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. If you want to export data to a file, use the **Export to File** functionality to run 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 the runs a **Cluster Computing** node. This gives you similar functionality as the distributed parametric sweep, but the results will be 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 Computing** node by right-clicking the study. Note that you need the **Advanced Study Options** enabled. 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 Sequence**. 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**. Note that all the resulting models will be 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.
- Create a parametric sweep that runs a solver and a class file that uses the Java API to modify the solution. This can be useful if you want COMSOL to communicate with another program, for instance. 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**.

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. The list includes both variables solved for and variables not solved for.

⚠ *Caution*

Computing this statistics requires a computation of the size of the assembled finite element model. This can take some time for large models.

6

Results Analysis and Plots

This chapter describes the features for visualization and analysis of the simulation results in COMSOL Multiphysics.

In this chapter:

Results

During results analysis and visualization there are these main operation features: *data sets*, *plots and plot groups*, *derived values and tables*, *exports*, and *reports*.

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. Go to Data Sets.

PLOT GROUPS AND PLOTTING

A plot group is a collection of plot features 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. Go to Plot Groups and Plotting.

Use a combination of data sets and plot groups to create cross-section plots. See Creating Cross-Section Plots in the *COMSOL Multiphysics User's Guide*.

DERIVED VALUES AND TABLES

Derived values define the evaluation of values of integrals, variables in points, or global variables. The evaluation results are stored in **Tables** and displayed in the **Results** window. Go to Derived Values and Tables.

EXPORTS

Export data, images, and animations from plot groups to files or use a player to visualize dynamic data. Go to Exports.

REPORTS

Create reports on various formats for documenting models as reports with text and images on an HTML format for easy viewing in a web browser.

VIEWS

To display the **Views** node under **Results**, click the **Show** button ( ) and select **Advanced Results Options**. Then go to User-Defined Views in the *COMSOL Multiphysics User's Guide* for information.

See Also

- Results Evaluation and Visualization in the *COMSOL Multiphysics User's Guide*
- Results API in the *COMSOL Java API Reference Guide*

Data Sets

See Table 6-1 in the section About Data Sets for links to all the data sets described in this section.

About Data Sets

Data sets contain or 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, create new data sets from other data sets. Add data sets to the **Data Sets** branch ( ) under **Results**.

All plots refer to data sets; the solutions are always available as the default data set. Characteristics of a data set include:

- 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.

> Note
>
> 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.

HOW TO ADD A DATA SET TO THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** ( ), right-click **Data Sets** ( ) and select an option from the context menu. Continue defining each data set as described. See Table 6-1 for links to all the data set types.

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 ( ), 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.

TABLE 6-1: DATA SET TYPES

LINK TO SECTION	ICON	DESCRIPTION AND PLOT USE
Average Evaluation or Integral Evaluation Data Sets	 ∫du	Creates a data set that computes the average of another data set, for example to plot the average.
Contour 2D Data Sets		To analyze on 2D contour lines. Use this data set for 2D arrow plots, 2D line plots, and 1D global plots.
Cut Line 2D or 3D Data Sets		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 Data Sets		Plots on cut planes are made on 3D data sets and can be visualized in either 2D or 3D plot groups.
Cut Point 1D, 2D, and 3D Data Sets	-●- ▣ 	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.
Edge 2D and 3D Data Sets	□ 	Plot and evaluate a value along an boundary (edge) in 2D or an edge in 3D. Use this data set to create plots for data on edges.
Function 1D, 2D, and 3D Data Sets	f(x)	Create a data set that can evaluate functions on a domain.
Isosurface 3D Data Sets		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 Data Sets		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.

TABLE 6-1: DATA SET TYPES

LINK TO SECTION	ICON	DESCRIPTION AND PLOT USE
Maximum Evaluation and Minimum Evaluation Data Sets		Creates a data set that computes the maximum or minimum of another data set.
Mesh Data Sets		To make a mesh available for visualization and results analysis. Use a Mesh feature in a plot group to get the actual plot.
Mirror 2D or Mirror 3D Data Sets		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.
Parameterized Curve 2D or 3D Data Sets		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 Data Sets		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.
Parametric Extrusion 1D or 2D Data Sets		To extend another data set by using a parameter, such as time, as a dimension.
Particle Data Sets		To visualize particle traces computed by a Particle Tracing interface.
Revolution 1D or 2D Data Sets		Use a revolution data set to visualize a 2D or 1D axisymmetric solution in 3D and 2D, respectively.
Solution Data Sets		To make solutions available for visualization and results analysis. Solvers create Solution data sets automatically.
Surface Data Sets		Plot and evaluate a value on surfaces (boundaries) in 3D. Use this data set to create plots for data on surfaces.

See Also

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- Defining Data Sets
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Expressions and Predefined Quantities
- Creating Cross-Section Plots
- Exporting Data from Data Sets

Average Evaluation or Integral Evaluation Data Sets

Select an **Average Evaluation** ( ) or **Integral Evaluation** ( ) data set, found under **Data Sets>Evaluation,** to compute the average or integral of another data set for example, to plot the average or integral.

Model

See Flow Past a Cylinder for an integral evaluation data set example, Model Library path **COMSOL_Multiphysics>Fluid_Dynamics>cylinder_flow**.

If you have the Batteries & Fuel Cells Module, see Edge Effects In a Spirally Wound Li-Ion Battery for an average evaluation data set example, Model Library path **Batteries_and_Fuel_Cells_Module>Batteries>li_battery_spiral_2d**.

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**. Summation is primarily relevant for reaction forces and is selected instead of integration for reaction forces when the automatic selection of integration method is active. Select the **Integration order** check box to edit or enter an integer value for the integration order (default: 4). Select a **Geometry level**—**Data set highest**, **Volume**, **Surface**, **Line**, or **Point**. The default is **Data set highest**, which means the highest geometry dimension for the data in the data set: typically volumes in 3D, surfaces in 2D, and lines in 1D.

Contour 2D Data Sets

Use a **Contour** ( ) data set 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.

DATA

Select a **Data set**. Only 2D data sets can be used.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

LEVELS

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 5). Otherwise, enter the values of the contour **Levels** or click the **Range** button ( ) to define specific range of levels.

Cut Line 2D or 3D Data Sets

Use a **Cut Line** data set in 2D ( ) or 3D ( ) to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis features available in 1D are available for cut line data sets as well as 3D plots and results analysis features for edges. These data sets are used to create cross-section line plots.

Model

For a 3D example, see Thin-Film Resistance: Model Library path **COMSOL_Multiphysics>Electromagnetics>thin_film_resistance.**

DATA

Select a **Data set**.

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). Select the **Bounded by points** check box to constrain the line between the defined points. If **Point and direction** is selected, enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D).

Cut Plane Data Sets

2D	For 2D models, select the **Additional parallel lines** check box to define multiple lines 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. 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.

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 features 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.

DATA

Select a **Data set**.

PLANE DATA

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, select a set of cut planes orthogonal to a coordinate axis and specify the transverse coordinate by entering the location along the transverse coordinate axis. If **General** is selected, select a **Plane entry method**—**Three points** or **Point and normal**. Enter **x**, **y** and **z** coordinates.

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. The **Distances** field refers to a direction that is normal to the cut plane.

ADVANCED

Select the **Define normal variables** check box to create variable names for the cut plane's normal vector. Under **Normal variables**, enter or edit the variable names for the components of the normal vector. Enter or edit the **nx**, **ny**, and **nz** variable names in the respective fields. The default names are `cpl1nx`, `cpl1ny`, and `cpl1nz`.

Cut Point 1D, 2D, and 3D Data Sets

Use a **Cut Point** data set in 1D ( ), 2D ( ), or 3D ( ) to plot and evaluate a value in a certain point along time or along a parametric solution and to create 1D, 2D, or 3D cross-sectional point plots. The choice of 1D, 2D, or 3D only controls the type of input the data set accepts. For example, a 1D cut point can only be added to 1D data sets, 2D can only be added to 2D data sets, and so forth. Either 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 **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.

DATA

Select a **Data set**.

POINT DATA

To specify the point data coordinates using the **Entry method** list:

- Select **Coordinates** (the default) to enter the coordinates. For 1D enter an **x** coordinate, for 2D enter **x** and **y** coordinates, and for 3D enter **x**, **y**, and **z** coordinates. Click the **Range** button ( ) to specify a range for the coordinate values using the **Range** dialog box.
- Select **From file** to read the coordinates from a text file. Enter the path and filename in the **Filename** field, or click **Browse** to browse to the file.
- Select **Grid** to enter grid coordinates for grid data in a grid. For 1D enter an **x** coordinate, for 2D enter **x** and **y** coordinates, and for 3D enter **x**, **y**, and **z**

coordinates. Click the **Range** button ( ) to specify a range for the coordinate values using the **Range** dialog box.

- Select Regular grid enter grid coordinates for a regular grid. Enter **Number of x points** (1D, 2D, and 3D), **Number of y points** (2D and 3D), and **Number of z points** (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 grid.

Edge 2D and 3D Data Sets

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.

DATA

Select a **Data set**.

SELECTION

Select the edges to include in the data set using the available selection tools or by selecting directly in the **Graphics** window.

Function 1D, 2D, and 3D Data Sets

Adding an **Function 1D**, **Function 2D**, or **Function 3D** node (f(x)), which is found under **Data Sets>Function,** 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).

Model

If you have the AC/DC Module, see Geoelectrics for an example of a 1D Function data set, Model Library path **ACDC_Module>General_Industrial_Applications>geoelectrics**.

FUNCTION

Select the function to evaluate on a domain.

PARAMETER BOUNDS

Available fields are based on the function dimension. Enter a **Name**. The **First, Second**, and **Third Parameter** default names are `s`, `t`, and `u` respectively. For each parameter, enter a **Minimum** lower bound and a **Maximum** upper bound 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 (1D), 100 (2D), and 30 (3D).

Isosurface 3D Data Sets

Use an **Isosurface** ( ) data set 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.

DATA

Select a **Data set**.

EXPRESSION

Enter an **Expression**, select a **Unit,** and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

LEVELS

Select an **Entry method** for the isosurfaces—**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 5). If **Levels** is selected, enter the values of the isosurface **Levels** or click the **Range** button ( ) to define specific range of levels.

Join Data Sets

Use an **Join** ( ) data set join 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. The Join data set has predefined methods to combine the solutions to get the 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. Click the **Go to Source** button ( ) to move to the selected solution data set node. 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 solution in a time-dependent, parametric, or eigenvalue solution from the list of solutions that appear underneath the **Solutions** list.

Note

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 you select one of the following method for combining the solution data sets:

- Select **Difference** (the default) to combine the data sets using a difference: `data1-data2`.
- 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. 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)`.
- 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 Evaluation and Minimum Evaluation Data Sets

Select the **Maximum** evaluation ( ) or **Minimum** evaluation ( ) data set, found under **Data Sets>Evaluation,** 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**—**Data set highest**, **Volume**, **Surface**, **Line**, or **Point**. The default is **Data set highest**, which means the highest geometry dimension for the data in the data set: typically volumes in 3D, surfaces in 2D, and lines in 1D.

Mesh Data Sets

Add a **Mesh** ( ) data set to make a mesh available for visualization and results analysis. Use a **Mesh** feature in a plot group to get the actual plot.

MESH

Select the **Mesh** to visualize from the list.

Mirror 2D or Mirror 3D Data Sets

2D

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.

3D

Use a **Mirror 3D** () data set to extend a solution defined on one side of a plane to the other side of a plane.

DATA

Select a **Data set**.

AXIS DATA

PLANE DATA

2D

For 2D models, select an **Axis entry method**—**Two points** or **Point and direction**.

- If **Two points** is selected, enter **Point 1** and **Point 2** for both the **x** and **y** coordinates.
- If **Point and direction** is selected, enter values for the **Point** and **Direction** for both the **x** and **y** coordinates.

3D	For 3D models, select a **Plane type**—**Quick** (the default) or **General**. If **Quick** is selected, select a **Plane**—**xy-planes**, **yz-planes**, or **zx-planes**. Enter the **z-coordinate**, **x-coordinate**, or **y-coordinate** based on the selection. • If **General** is selected, select a **Plane entry method**—**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. Then if **Three points** is selected, enter coordinates in the **Point 1**, **Point 2**, and **Point 3** fields for **x**, **y**, and **z** coordinates (SI unit: m). • If **Point and Normal** is selected, enter **Point** and **Normal** vector coordinates for **x**, **y**, and **z** coordinates (SI unit: m).

ADVANCED

Select the **Define variable** check box to create a **Positive side indicator** variable. The default variable, `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.

Parameterized Curve 2D or 3D Data Sets

Use a **Parameterized Curve** data set in 2D ( ) or 3D ( ) 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.

DATA

Select a **Data set**.

PARAMETER

Enter a **Name** and the **Minimum** and **Maximum** range of the parameter curve.

EXPRESSIONS

Enter functions for the coordinates of the parameter. For 2D enter values in the **x** and **y** fields. For 3D, enter values in the **x**, **y**, and **z** fields.

RESOLUTION

Enter the number of subdivisions of the parameter range. The default is 1000.

Parameterized Surface Data Sets

Use a **Parameterized Surface** ( ) data set 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.

DATA

Select a **Data set**.

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 based on the model dimension.

RESOLUTION

Enter the number of subdivisions of the parameter ranges. the default is 200.

Parametric Extrusion 1D or 2D Data Sets

The **Parametric Extrusion** data set in 1D ( ) and 2D ( ) extends another data set by using a parameter, such as time, as a dimension.

Model

See The KdV Equation and Solitons for an example of a 1D parametric extrusion data set: Model Library path **COMSOL_Multiphysics>Equation-Based_Models>kdv_equation**.

DATA

Select a **Data set**.

SETTINGS

Select the **Level scale factor** check box to edit the field. The **Separate levels** check box is selected by default.

Particle Data Sets

Note

Particle data sets require the Particle Tracing Module.

Use a **Particle** (⁘) data set 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, provided that the **Generate default plots** option is selected in the **Study**. Selections may be added to the particle data set which allows, for example, the number or fraction of particles in a given domain or on a given boundary to be computed during results processing. Visualize the particles using a plot of the particle trajectories in the original dimension, as a Poincaré map, and 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_<tag>` where `<tag>` is the interface identifier for the specific Particle Tracing interface.

In the 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 model identifier prepended. For example, for a 3D model, with tag `<mod1>`, and **Dependent Variables** `qx`, `qy`, and `qz` for the interface, the correct expression to enter is `mod1.qx`, `mod1.qy`, and `mod1.qz`.

Revolution 1D or 2D Data Sets

1D Axi

Use a **Revolution** 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.

2D

Use a **Revolution** 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.

Model

See Tubular Reactor for a 2D example: Model Library path **COMSOL_Multiphysics>Chemical_Engineering>tubular_reactor**.

DATA

Select a **Data set**.

AXIS DATA

Here you specify the revolution axis or point by a method based on the space dimension:

- In 1D, enter a value in the x field to specify the revolution point.
- In 2D, select **Two Points** or **Point and direction** from the **Axis entry method** list, and then enter the coordinates for the two points or the coordinates for the base point and the direction vector.

REVOLUTION LAYERS

1D

For 1D models, enter a value in the **x** field to specify the revolution point.

2D

For 2D models, 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 revolution 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.
- If **Point and Direction** is selected, enter **Point** and **Direction** vector for **x** and **y** coordinates.

In the **Layers** field, choose **Fine**, **Normal** (the default), **Coarse**, or **Custom** from the **Number of layers** list. 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 subdivisions about the revolution axis (default value: 50).

Enter the **Start angle** (in degrees) for the revolved model (default value: 0 degrees).

In the **Revolution angle** field, enter the angle (in degrees) to revolve the model to see into the geometry in degrees (default value: 360 degrees; that is, a full revolution). Use 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

From the **Map plane to** list, select which plane to map the axisymmetric solution to—select **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.

Select the **Define variables** check box to create variable names in the revolved geometry. Under **Space variables**, enter or edit the variable names for the revolved coordinate system. Enter or edit the **x**, **y**, and **z** variable names in the respective fields. Under **phi**, enter or edit the variable name for **phi** in its field. Phi is the name of the angle variable in the revolved coordinate system. The default names are `rev1x`, `rev1y`, `rev1z`, and `rev1phi`.

Solution Data Sets

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

- Select a **Solution** to make available for visualization and results analysis.
- If there is more than one model, select the geometry to perform visualization and results analysis on by selecting the corresponding model from the **Model** list.
- Select the **Frame** to evaluate the coordinates in—**Mesh**, **Material**, 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 (in degrees) in the **Solution at angle (phase)** field (default: 0 degrees) to evaluate 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 value in the **Scale factor** field (default: 1; that is, no scaling) to multiply the solution by a real-valued scale factor.

Surface Data Sets

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.

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** is the default setting, which 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.
- **xy-plane** is the local *x*- and *y*-axes representing the global *yx*-plane.
- **yz-plane** is the local *x*- and *y*-axes representing the global *zy*-plane.

- **zx-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.

SELECTION

Select the boundaries to include in the data set using the available selection tools or by selecting directly in the **Graphics** window.

Plot Groups and Plotting

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. Define plot groups for 1D, 2D, and 3D plots and then create individual or a series of plots in a plot group. Use the information in the form of data and images to generate a report or export to a file.

Attributes can also be added to a plot to modify the plot's behavior—**Deformation** ( ) attributes deform a plot, **Color Expression** ( ) attributes modify the color of a plot, and filtering to only include parts of the plot is selected using the **Filter** attribute ( ). Select an appropriate color table for the plots' color expressions based on your audience and what you plan to do with the final analysis.

These main topics are included in this section. Links to individual plots are found in Table 6-2 and Table 6-3.

- 1D Plot Groups
- Polar Plot Groups
- 3D Plot Groups and 2D Plot Groups
- Color Expressions
- Deformations
- Filters
- Height Expressions

See Also

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Creating Cross-Section Plots
- Combining Plots
- Working with Geometry

1D Plot Groups

Important

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.

Note

At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot.

The time-related settings only display on the interfaces for time-dependent models.

Use a **1D Plot Group** ( ) to plot these graphs: Global Plots, Histogram, Line Graphs, Nyquist Plot, Point Graphs, or Table Graph. 1D plot groups are also used to create cross-section plots.

DATA

Select a **Data set**. 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.

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 **Data set** selected. If **Interpolated times** is selected, enter **Times** or click the **Range** button ( ) to select and define specific times. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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 **x minimum**, **x maximum**, **y minimum**, and **y maximum** limits already assigned based on the data set.

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 predefined **x spacing** and **y spacing** fields. Also specify extra grid points on *x*-axis and *y*-axis in the **Extra x** and **Extra y** fields. Click the **Range** button ( ) next to each field to specify a range of values.

LEGEND

Specify the position of the legends for the plots in the plot group. From the **Position** list, select **Upper right** (the default), **Lower right**, **Upper left**, 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

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.

ADDING PLOTS TO THE 1D PLOT GROUP IN THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** ( ), right-click the **1D Plot Group** ( ) node and select an option from the context menu such as **Line Graph** or **Point Graph**. Continue defining each plot as described (see Table 6-2).

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 windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

TABLE 6-2: 1D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION
Global Plots	📈	Plot a global scalar quantity as a function of time or a parameter.
Histogram	📊	Plot a histogram showing the distribution of a quantity.
Line Graphs	📈	Plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line.
Nyquist Plot	○	Display a Nyquist plot, which shows the magnitude and phase of a frequency response.
Point Graphs	📈	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.
Table Graph	▦	Table plots display data from a table with one line per output column.
*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.

* Cross-section plots are not selected from this menu. Instead, these are created it in two steps using data sets and plot groups to generate the plot.

See Also

- Polar Plot Groups
- 3D Plot Groups and 2D Plot Groups

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- Entering Ranges and Vector-Valued Expressions
- Solver Studies and Study Types
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Combining Plots
- Creating Cross-Section Plots

Polar Plot Groups

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. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

AXIS

Select the **Manual axis limits** check box to edit the **r minimum** and **r maximum** already assigned based on the current plot.

GRID

Select the **Manual spacing** check box to edit the predefined **r spacing** and θ **spacing** (SI unit: degrees).

Also specify extra grid points in the **Extra** θ and **Extra r** fields. Click the **Range** button ( ) next to each field to specify a range of values.

LEGEND

Specify the position of the legends for the plots in the plot group. From the **Position** list, select **Upper right** (the default), **Lower right**, **Upper left**, 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 setting, but also select any other plot window, 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.

See Also

- 1D Plot Groups

In the *COMSOL Multiphysics User's Guide*:

- Defining a 1D Plot Group and Adding Plots
- Entering Ranges and Vector-Valued Expressions
- Combining Plots

3D Plot Groups and 2D Plot Groups

Important

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.

Note

At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot.

The time-related settings only display on the interfaces for time-dependent models.

Use a **2D Plot Group** ( ) to combine several 2D plots into one to visualize all the plots simultaneously. Use a **3D Plot Group** ( ) to combine several 3D plots into one to visualize all the plots simultaneously. For example, combine a surface plot and a streamline plot using the same data set.

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

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

PLOT SETTINGS

Select a **View**. The default is **Automatic**. Define other **Views** under **Definitions**. Similarly, manually enter axis labels by selecting the **x-axis label** and **y-axis label** check boxes. This can be useful for scatter plots, for example, where the axes represent other quantities than the *x* and *y* directions. 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 define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Select a **Frame**—**Material** (the default), **Mesh**, **Geometry frame**, or **Spatial**.

I

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).

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 create a new plot window for the plots in this plot group. 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.

ADDING PLOTS TO 2D AND 3D PLOT GROUPS IN THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** ( ), right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) node and select one or more options from the context menu. Each plot group can have several plots combined to create a meaningful representation of the data. Continue defining each plot as described (see Table 6-3).

Also add attributes to 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 2D Line and 2D Surface plots.

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 in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Arrow Line Plots	3D 2D	Plot a vector quantity as arrows on lines or edges (3D).	Color Expression, Deformation, and Filter
Arrow Surface Plots	3D 2D	Visualize a vector quantity in arrows.	Color Expression, Deformation, and Filter
Arrow Volume Plots		Visualize a vector quantity as arrows in a volume.	Color Expression, Deformation, and Filter

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Contour Plots	3D 2D	Visualize a scalar quantity as a contour plot.	Color Expression, Deformation, and Filter
Coordinate System Volume, Surface, and Line Plots	3D 2D	To plot coordinate systems for 2D and 3D models. Found on the More Plots menu.	Deformation (2D surface and 3D volume) Deformation and Filter (2D Line, 3D surface, and 3D line)
Isosurface Plots		Plot a scalar quantity as an isosurface plot.	Color Expression, Deformation, and Filter
Line Plots	3D 2D	Plot a quantity on lines, boundaries (2D), or edges (3D).	Deformation, Filter, and Height Expression (2D only)
Max/Min Volume, Max/Min Surface, and Max/Min Line Plots		To plot the maximum and minimum of an expression and the points there they are attained within the geometry. Found on the More Plots menu. and select from the following: Max/Min Volume (), Max/Min Surface 2D (), Max/Min Surface 3D (), Max/Min Line 2D (), and Max/Min Line 3D ()	Deformation
Mesh Plots		To display a mesh.	Deformation and Filter
Multislice Plots		Display a scalar quantity on slices in multiple directions inside a 3D domain. Found on the More Plots menu.	Deformation and Filter

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Particle Tracing Plots	3D 2D	Visualize the trajectory of a massless particle subject to a flow field. Found on the More Plots menu.	Color Expression and Deformation
Particle Tracing with Mass Plots	3D 2D	Make a particle tracing plot considering the particle’s mass. Found on the More Plots menu.	Color Expression and Deformation
Particle Trajectories Plots	3D 2D	Visualize the trajectory of a massless particle subject to a flow field in 2D or 3D.	Color Expression and Deformation
Phase Portrait Plots		Visualize particle traces using a phase portrait in 2D. Found under the More Plots menu.	Color Expression
Poincaré Map Plots	3D 2D	Visualize particle traces as a Poincaré map in 2D, 2D axial symmetry, and 3D. Found on the More Plots menu.	Color Expression
Principal Stress Volume Plots		Plot principal stress and strain in structural mechanics models. Found on the More Plots menu.	Color Expression, Deformation, and Filter
Principal Stress Surface Plots	3D 2D	Plot principal stress and strain in structural mechanics models. Found on the More Plots menu.	Color Expression, Deformation, and Filter

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Scatter Surface and Scatter Volume Plots	▦ ▣	Scatter surface plots, found on the More Plots menu, visualize a scalar quantity as scattered spheres or disks. The radius of the spheres can be proportional to the value of a quantity. 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
Slice Plots	▥	Display a scalar quantity on slices inside a 3D domain.	Deformation and Filter
Streamline Plots	3D ≋ 2D ≋	Plot a vector field as a streamline plot.	Color Expression and Deformation
Surface Plots	3D □ 2D ■	Display a quantity on a 2D domain or a 3D boundary.	Deformation, Filter, and Height Expression (2D plots)
Table Surface Plots	■	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
Volume Plots	■	To display a quantity inside a 3D domain.	Deformation and Filter

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
PLOT ATTRIBUTES			**ADD TO THESE PLOTS**
Color Expressions		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
Deformations		Deform the plot according to a vector quantity, for example, the displacement field in structural mechanics.	All plots except scatter plots.
Filters		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 Expressions		Height attributes introduce a 3D height on a 2D surface plot.	2D line plot, 2D surface plot
CROSS SECTION PLOTS			
*1D, 2D, and 3D Cross-Section Point Plots		View an expression at an arbitrary set of spatial coordinates.	
*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.	

TABLE 6-3: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
*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.

See Also

- 1D Plot Groups
- Polar Plot Groups

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Combining Plots
- Creating Cross-Section Plots
- User-Defined Views
- Solver Studies and Study Types

Arrow Line Plots

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** ( ). 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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set** from the list. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter or select the **x component** and **y component** for 2D, the **r component** and **z component** for 2D axial symmetry, or the **x component**, **y component**, and **z component** for 3D. Click the **Replace Expression** (▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

COLORING AND STYLE

Select an **Arrow type**—**Arrow** (the default) or **Cone**.

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 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.

Select the position of the **Arrow base**—**Tail** (the default) or **Head**.

For manual scaling of the arrows, select the check box and enter a **Scale factor** for the arrows using a positive scalar number in the field or by using the associated slider.

From the **Placement** list, select **Uniform** (the default) for arrows positioned uniformly on the surface, or select **Elements** for arrows positioned in the mesh elements (that is, more densely placed arrows where the mesh density is high).

Specify the total **Number of arrows** (default: 200 arrows).

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.

INHERIT STYLE

Use this functionality to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Arrow Surface Plots

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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set** from the list. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter or select **x component** and **y component** for 2D, **r component** and **z component** for 2D axial symmetry, or **x component**, **y component**, and **z component** for 3D. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

ARROW POSITIONING

Note

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.

In the **x grid points** and **y grid points** fields (**r grid points** and **z grid points** in 2D axial symmetry), select an **Entry method**. If **Number of points** is selected, enter the number of **Points** in each direction (default: 15). If **Coordinates** is selected, enter **Coordinates** in the field or click the **Range** button ( ) to define specific coordinates.

COLORING AND STYLE

Select an **Arrow type**—**Arrow** (the default) or **Cone**.

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 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.

Select the position of the **Arrow base**—**Tail** (the default) or **Head**.

For manual scaling of the arrows, select the check box and enter a **Scale factor** for the arrows using a positive scalar number in the field or by using the associated slider.

From the **Placement** list, select **Uniform** (the default) for arrows positioned uniformly on the surface, or select **Elements** for arrows positioned in the mesh elements (that is, more densely placed arrows where the mesh density is high).

Specify the total **Number of arrows** (default: 200 arrows)

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.

INHERIT STYLE

Use this functionality to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Arrow Volume Plots

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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set** from the list. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter or select **x components**, **y components**, and **z components**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

ARROW POSITIONING

In the **x grid points**, **y grid points**, and **z grid points** fields, select an **Entry method**. If **Number of points** is selected, enter the number of **Points** in each direction (default: 7). If **Coordinates** is selected, enter **Coordinates** or click the **Range** button ( ) to define specific coordinates.

COLORING AND STYLE

Select an **Arrow type**—**Arrow** (the default) or **Cone**.

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 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.

Select the position of the **Arrow base**—**Tail** (the default) or **Head**.

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 3).

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.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Contour Plots

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**, or **Filter** subnodes as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit,** and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

LEVELS

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 5). Otherwise, enter the values of the contour **Levels** or click the **Range** button ( ) to define specific range of levels.

COLORING AND STYLE

Select a **Contour type**—**Lines** or **Filled**. If **Lines** is selected, also select the **Level labels** check box to display line labels on the graph.

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.

Select a **Coloring** for the contours—**Color table** (default) or **Uniform**.

- If **Color table** is selected, choose a **Color table** from the list (default in most plots: Rainbow). The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.
- If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. A higher resolution means that elements are split into smaller patches during rendering. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. Select an option from the **Smoothing** list—**None**, **Internal** (the default), or **Everywhere**. The **Recover** default is **Off** because accurate derivative recovery takes processing time. Or select another option from the **Recover** list—**Within domains** or **Everywhere**.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Coordinate System Volume, Surface, and Line Plots

Add Coordinate System plots in 3D () or 2D (). Use the **Coordinate System Volume** (3D), **Coordinate System Surface** (2D and 3D), and **Coordinate System Line** (3D) plots to plot coordinate systems. These plots are located on the context menu under **More Plots**. Right-click to add a **Deformation** or **Filter** (for 2D and 3D Coordinate System Line and 3D Coordinate System Surface plots), if required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** means that the plot uses the same data set as the plot group it belongs to. For **Parametric Solver** and **Parametric Sweep** studies, select an option from the **Parameter values** list. For **Parametric Sweep**, also select a **Time**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

COORDINATE SYSTEM

Select a **Coordinate System** to plot. By default the Global coordinate system is selected and the list contains any additional coordinate systems that the model includes.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

POSITIONING

For **3D Plot Groups** - **Coordinate System Volume** plots or **2D Plot Groups - Coordinate System Surface** plots: In the **x grid points**, **y grid points**, and **z grid points** (3D only) fields select an **Entry method**. If **Number of points** is selected, enter the number of **Points** in each direction (the default is 15). If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to select and define specific coordinates.

COLORING AND STYLE

Select an **Arrow length**—**Proportional** (the default), or **Normalized**. Select the position of the **Arrow base**—**Tail** (the default) or **Head**. Enter a **Scale factor**.

For **2D** and **3D Plot Groups** - **Coordinate System Line** and **3D Plot Groups** - **Coordinate System Surface** plots, specify the total **Number of arrows** (default: 200 arrows). Use the **Displayed fraction** field or the associated slider to control the fraction (0–1) of the arrows to display (the default value is 1; that is, all arrows).

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Global Plots

Use a **Global** ( ) plot to graph a scalar quantity as a function of time or a parameter. Add these to a **1D Plot Group** or **Polar Plot Group**.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to. For **Parametric Solver** and **Parametric Sweep** studies, select an option from the **Parameter values** list. For **Parametric Sweep**, also select a **Time**.

Y-AXIS DATA (1D PLOTS) OR R-AXIS DATA (POLAR PLOTS)

See Also

- Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings in the *COMSOL Multiphysics User's Guide*

Enter an **Expression** and **Description** in the table or field. Click the **Replace Expression** ( ) or **Add Expression** ( ) buttons to select predefined expressions based on the physics of the model. Use the **Move Up** ( ), **Move Down** ( ), and **Delete** ( ) buttons as required to move rows in the table.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

X-AXIS DATA OR θ ANGLE DATA (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. 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.

Tip

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.

From the **Parameter** list, select—**Parameter value**, **Time**, **Solution number**, **Frequency spectrum** or **Expression**.

- If **Parameter value**, **Time**, or **Solution number** is selected, the *x*-axis represents the parameter values (for a parametric study), time (for a time-dependent study), or solution numbers (for an eigenvalue or modal study, or for a parametric solution with more than one parameter), respectively.
- If **Frequency spectrum** is selected, COMSOL Multiphysics computes 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 ($u_{\max}/\sqrt{2}$).
- If **Expression** is selected, click the **Replace Expression** (⇄ ▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select a **Unit** and select the check box to enter a **Description** (or edit the default).

COLORING AND STYLE

Select a **Line style** from the **Line** list—**None**, **Cycle**, **Solid** (the default), **Dotted**, **Dashed**, or **Dash-dot**. Select **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. In the **Line markers** section, select a **Marker**—**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) 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).

LEGENDS

The **Show legends** check box is selected by default to display to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, these values also display. For global plots, when **Automatic** is selected from the **Legends** list (the default), select or clear the **Expression** and **Description** check boxes to control what to include in the automatic legends (by default they include the description only). If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

Histogram

Use a **Histogram** (📊) to plot a histogram that shows how a quantity is distributed over the geometry (mesh volume). 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. Also view the histogram as a plot showing the area in-between contours or isosurfaces. The histogram can be normalized and also displayed as a cumulative plot, and it can appear as a discrete or a continuous function. Add **Histogram** plots to a **1D Plot Group**.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. Select **From parent**, a **Solution**, **Cut Line**, or **Parameterized Curve** data set, or **None**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a compatible **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** (➕) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit or replace the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

BINS

Use the **Entry method** list to define the bins for the histogram's *x*-axis. Select **Number of bins** (the default) to specify the number of bins (default value: 10 bins), or select **Limits** to specify a range of limits (`1 2 3 4`, for example) for the histogram bins. Click the **Range** button ( ) to define a range using the **Range** dialog box.

OUTPUT

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 select **Discrete** to plot it as a discrete function (that is, using a constant level in each bin).

From the **Normalization** list, select one of the following normalizations:

- Select **Integral** to normalize the histogram so that the integral is equal to 1.
- Select **None** (the default) to show the actual element volume without any normalization.
- Select **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.

COLORING AND STYLE

Under **Line style**, select a line style from the **Line** list—**None**, **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. Select a **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Use the **Width** slider or field to control the line width. In the **Line markers** section, select a **Marker**—**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) 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).

Note

The line markers are only available for histograms using a continuous function.

LEGENDS

The **Show legends** check box is selected by default to display to the right of the plot. When **Automatic** is selected from the **Legends** list (the default), the program defines the legend text. Select **Manual** from the **Legends** list to enter your own legend text into the table.

QUALITY

Select a plot **Resolution**—**Extra fine, Finer**, **Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. The **Recover** default is **Off** because accurate derivative recovery takes processing time. If required, select another option from the **Recover** list—**Within domains** or **Everywhere**.

Isosurface Plots

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 **Color Expression**, **Deformation**, or **Filter** subnodes as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** (+) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

LEVELS

Select an **Entry method** for the isosurfaces—**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 5). If **Levels** is selected, enter the values of the isosurface **Levels** or click the **Range** button ( ) to define specific range of levels. Before plotting, select the **Interactive** check box to move the isosurfaces interactively using the slider or by typing a shift in the **Shift** field. A zero shift represents the original position of the isosurfaces.

Note

In interactive mode, the legend covers the whole range of the field over all elements, but in noninteractive mode, only the range of the isolevels is covered.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Select a **Color table**. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

QUALITY

! *Important*	**Custom** refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer**, **Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field.

The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to specify where the recovery applies—**Within domains** or **Everywhere**. By default the **Use derivatives** check box is selected. To increase the speed of computation, click to clear the check box. However, this results in less smooth surfaces.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Line Graphs

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 these to a **1D Plot Group** or a **Polar Plot Group**.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. Select **From parent**, a **Solution**, **Cut Line**, or **Parameterized Curve** data set, or **None**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. For time-dependent, parametric, or eigenvalue/ eigenfrequency solutions, also select times, parameters, or eigenvalues/ eigenfrequencies.

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.

Y-AXIS DATA OR R-AXIS DATA (POLAR PLOTS)

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

X-AXIS DATA OR θ-ANGLE DATA

Note

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** from the **Parameter** list to visualize along an arc length, or select **Expression** to visualize along, for example, a coordinate expression.

If **Expression** is selected, enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** (▸▾) or **Insert Expression** (+ ▾) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

COLORING AND STYLE

Select a **Line style** from the **Line** list—**None**, **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. Select a **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. In the **Line markers** section, select a **Marker**—**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) 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).

LEGENDS

If required, select the **Show legends** check box to display a legend to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, these values also display. For point plots, the legend displays the coordinate (or vertex number). If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer**, **Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. The **Recover** default is **Off** because accurate derivative recovery takes processing time. If required, select another option from the **Recover** list—**Within domains** or **Everywhere**.

Line Plots

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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

RANGE

As required, 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.

COLORING AND STYLE

Select a **Line type**—**Line** or **Tube**. If **Tube** is selected, enter a **Tube radius expression** (the radius of the tube). Select the **Radius scale factor** check box to enter a number (SI unit: m). Select a **Coloring**—**Color table** (default) or **Uniform**. If the default **Color table** (**Rainbow** in most plots) is not suitable for the plot, try other options. If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

QUALITY

Important	**Custom** refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**. The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to specify where the recovery applies—**Within domains** or **Everywhere**.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, **Deform scale factor**, and **Tube radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Max/Min Volume, Max/Min Surface, and Max/Min Line Plots

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. Add a **Deformation** subnode as required. These plots are located under **More Plots**. When plotting the maximum and minimum value, and associated table appears in a **Results** window (underneath the plot with the default COMSOL Desktop layout). The table contains the maximum and minimum values along with the coordinates for the corresponding locations.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select the predefined expression to compute the maximum and minimum for. Or enter an **Expression**. Select a **Unit** to plot. To enter a **Description** (or edit the default), select the check box.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

ADVANCED

Define the number of refinements of each mesh element when calculating the maximum and minimum by entering a value in the **Element refinement** field (the default is 2). Enter a **Display Precision** for the number of decimals displayed in the labels. The default is 6.

The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to specify where the recovery applies—**Within domains** or **Everywhere**.

COLORING AND STYLE

In this section you can select a color for the maximum and minimum markers from the **Color** list: **Custom**, **Black** (the default), **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to select a color from the **Custom color** palette that opens.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Deform scale factor** check box is selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Mesh Plots

Use a **Mesh** plot to display a mesh in 2D or 3D. However, the mesh can be visualized without solving a model using a mesh data set. Add **Deformation** or **Filter** subnodes as required.

> *Tip*
>
> A Mesh plot can also be created by right-clicking the **Mesh** node and selecting **Plot** ( ).
>
> Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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**.

3D only: If **Volume** is selected, select the base **Element type** to visualize—**All**, **Tetrahedron**, **Prism**, or **Hex**.

COLOR

From the **Element color** list, select any basic color, **Quality** (the default), **Custom**, or **None**. Select a **Color table** for the element quality. From the **Wireframe color** list, select any basic color, **Custom**, or **None**.

ELEMENT FILTER

Specify the elements to display. If required, select the **Enable filter** check box and select a **Criterion**— **Logical expression** (the default), **Random**, **Worst Quality**, **Best Quality**, **Size**, or **Expression**.

- If **Expression** or **Logical expression** is selected, enter an **Expression** in the field.
- For all **Criterion** (except **Logical expression**), enter the **Fraction** of elements to show (the default is 1).

SHRINK ELEMENTS

Enter an **Element scale factor** between 0 and 1 to scale elements in the mesh plot.

Multislice Plots

Use a **Multislice Plot** ( ) to display a scalar quantity on slices in multiple directions inside a 3D domain. You find this plot on the **More Plots** submenu. Add **Deformation** and **Filter** subnodes as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

MULTIPLANE DATA

Specify the slices for the **x-planes**, **y-planes**, and **z-planes**. Select an **Entry method**—**Number of planes** or **Coordinates**. If **Number of planes** is selected, enter the number of **Planes**. If **Coordinates** is selected, enter the applicable grid **Coordinates** or click the **Range** button ( ) to define a range of values.

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.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Select a **Color table**. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**.

The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to control where the recovery operation applies—**Within domains** or **Everywhere**.

INHERIT STYLE

Use these settings to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes from that plot. The **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default to inherit those style attributes. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Nyquist Plot

Use a **Nyquist Plot** ( ) 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 **Nyquist Plot** nodes to a **Polar Plot Group** or a **1D Plot Group**.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to. For **Parametric Solver** and **Parametric Sweep** studies, select an option from the **Parameter values** list. For **Parametric Sweep**, also select a **Time**.

EXPRESSIONS

In the table, enter an **Expression**, select a compatible **Unit**, and enter or edit a **Description**. Use a new row in the table for each expression. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Use the **Move Up** ( ), **Move Down** ( ), and **Delete** ( ) buttons as required to move rows in the table.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

COLORING AND STYLE

To include a unit circle in the Nyquist plot, select the **Show unit circle** check box. Under **Line style**, select a line style from the **Line** list—**None**, **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. Select a **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Use the **Width** slider or field to control the line width. In the **Line markers** section, select a **Marker**—**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) 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).

LEGENDS

The **Show legends** check box is selected by default to display to the right of the plot. When **Automatic** is selected from the **Legends** list (the default), the program defines the legend text. Select or clear the **Expression** and **Description** check boxes to control what to include in the automatic legends (by default they include the description only). Select **Manual** from the **Legends** list to enter your own legend text into the table.

See Also

- Global Plots in the *COMSOL Multiphysics User's Guide*:

Particle Tracing Plots

Note

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 ( ). This plot is found under the **More Plots** submenu. 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** as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Note

Go to Particle Tracing Plots in the *COMSOL Multiphysics User's Guide* for detailed instructions.

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter or select **x component** and **y component** for 2D, **r component** and **z component** for 2D axial symmetry, or **x component**, **y component**, and **z component** for 3D. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

PARTICLE POSITIONING

Under **Particle Positioning**, for *3D* only, enter the initial position of particles in the **x**, **y**, and **z** fields. Click the **Range** button ( ) to define a range using the **Range** window.

For *2D*, 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 or click the **Range** button ( ) to define a range using the **Range** window.

- If **Boundary coordinates** is selected, select an item from the **Named selection** list and select an **Entry method**—**Number of points** or **Boundary parameters**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.
  - 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** or click the **Range** button ( ) to define a range using the **Range** window.

RELEASE

Specify when to **Release particles**—**At times**, **Once** (the default), or **At intervals**. If **At intervals** is selected, enter a **Time between releases**. If **At times** is selected, enter multiple **Times** or click the **Range** button ( ) to define a range using the **Range** window. To delay a release, select the **Start time** check box and enter a time.

COLORING AND STYLE

Select a **Line style**—**Line**, **Tube**, or **None**. If **Tube** is selected, enter a **Tube radius expression** for the radius of the tube (SI unit: m); the default is 1 m. Select the **Radius scale factor** check box to enter a number. If there is no **Color Expression** subnode that determines the color of the lines, select a line **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Select a **Point style**—**Point** or **None**. If **Point** is selected, enter a **Point radius**. To edit the **Radius scale factor** select the check box and enter a number. Select a point **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Select a **Point motion** to specify what should happen as particles leave the 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.

If the plan is to create an **Animation** report, and if required, select the **Plot along lines when animating** check box.

QUALITY

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, or **Coarser**. The **Recover** default is **Off** because accurate derivative recovery takes processing time. If required, select another option from the **Recover** list—**Within domains** or **Everywhere**. In the **ODE solver settings** section, enter or select a **Relative tolerance, Absolute tolerance**, and **Step size**.

ADVANCED

Note

The **Advanced** section contains settings that do not normally need to be adjusted.

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**.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, **Color and data range**, **Deform scale factor, Point radius scale factor**, and **Tube radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Particle Tracing with Mass Plots

Note

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 with Mass** plot in 2D ( ) or 3D ( ) to visualize the trajectory of a particle with mass and subject to a flow field. This plot is found under the **More Plots** submenu. Add a **Color Expression** or **Deformation** as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Note

Go to Particle Tracing with Mass Plots for instructions and background information as this plot type has a lot of detail.

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

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**.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

PARTICLE POSITIONING

Under **Particle Positioning**, for *3D* only, enter the initial position of particles in the **x**, **y**, and **z** fields. Click the **Range** button ( ) to define a range using the **Range** window.

For *2D*, 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 or click the **Range** button ( ) to define a range using the **Range** window.
- If **Boundary coordinates** is selected, select an item from the **Named selection** list and select an **Entry method**—**Number of points** or **Boundary parameters**. Click the **Go to**

Source button ( ) to move to the node that the selection in the list next to the button refers to.

- 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** or click the **Range** button ( ) to define a range using the **Range** window.

RELEASE

Specify when to **Release particles**—**At times**, **Once** (the default), or **At intervals**. If **At intervals** is selected, enter a **Time between releases**. If **At times** is selected, enter multiple **Times** or click the **Range** button ( ) to define a range using the **Range** window. To delay a release, select the **Start time** check box and enter a time.

COLORING AND STYLE

Select a **Line style**—**Line**, **Tube**, or **None**. If **Tube** is selected, enter a **Tube radius expression** for the radius of the tube (SI unit: m); the default is 1 m. Select the **Radius scale factor** check box to enter a number. If there is no **Color Expression** subnode that determines the color of the lines, select a line **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Select a **Point style**—**Point** or **None**. If **Point** is selected, enter a **Point radius**. To edit the **Radius scale factor** select the check box and enter a number. Select a point **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Select a **Point motion** to specify what should happen as particles leave the 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.

If the plan is to create an **Animation** report, and if required, select the **Plot along lines when animating** check box.

QUALITY

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, or **Coarser**. The **Recover** default is **Off** because accurate derivative recovery takes processing time. If required, select another option from the **Recover** list—**Within domains** or **Everywhere**. In the **ODE**

solver settings section, enter or select a **Relative tolerance, Absolute tolerance**, and **Step size**.

Note

The **Advanced** section contains settings that do not normally need to be adjusted.

ADVANCED

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**. In the **Particle velocity variables** section, 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**).

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, **Color and data range**, **Deform scale factor, Point radius scale factor**, and **Tube radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Particle Trajectories Plots

Note

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 since 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** or **Deformation** as required.

Tip	Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Model	• Particle Trajectories in a Laminar Static Mixer: Model Library path **Particle_Tracing_Module/Fluid_Flow/laminar_mixer_particle** • Ion Cyclotron Motion: Model Library path: **Particle_Tracing_Module/Tutorials/ion_cyclotron_motion**

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

COLORING AND STYLE

Select a **Line style**—**Line**, **Tube**, or **None**. If **Tube** is selected, enter a **Tube radius expression** for the radius of the tube (SI unit: m); the default is 1 m. Select the **Radius scale factor** check box to enter a number. If there is no **Color Expression** subnode that determines the color of the lines, select a line **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Select a **Point style**—**Point** or **None**. If **Point** is selected, enter a **Point radius**. To edit the **Radius scale factor** select the check box and enter a number. Select a point **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

If the plan is to create an **Animation** report, and if required, select the **Plot along lines when animating** check box.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The

Color, **Color and data range**, **Deform scale factor, Point radius scale factor**, and **Tube and ribbon radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Phase Portrait Plots

Note	This plot is available with the Particle Tracing Module.

Use a **Phase Portrait** plot ( ), which is added from the **More Plots** menu, 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 x-y plane represents a particle. By default, the position is taken as the distance from the origin (0,0,0) for 3D models.

2D 2D Axi	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 3D Plot Groups and 2D Plot Groups page under **Plot Settings**. Then in the **Graphics** window click the **Zoom extents** button ( ) to see the phase portrait.

Tip	Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Model	For an example, see Ideal Cloak: Model Library path **Particle_Tracing_Module/Ray_Tracing/ideal_cloak**.

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

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).

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

COLORING AND STYLE

Enter a **Point radius**. Select the **Radius scale factor** check box to enter a number to scale the value more than the default of 1. Select a **Color** for the point.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. To prevent a style attribute from being inherited, click to clear the **Color**, **Color and data range**, or **Point radius scale factor** check boxes, or select **None** from the **Plot** list.

Poincaré Map Plots

3D

Note

This plot is available with the Particle Tracing Module and with 3D models only.

For 3D models, use a **Poincaré Map** plot, which is added from the **More Plots** menu, to 2D ( ) and 3D ( ) plot groups, to visualize particle trajectories using a *Poincaré map* (sometimes called a *first recurrence map*). This type of plot is useful for visualizing 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 is constructed by first defining a **Cut Plane** on the **Particle** data set. The Poincaré map parent plot group should point to this cut plane. 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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Model

Particle Trajectories in a Laminar Static Mixer: Model Library path **Particle_Tracing_Module/Fluid_Flow/laminar_mixer_particle**

DATA

Select a **Cut plane**. **From parent** 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 node that the selection in the list next to the button refers to.

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).

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

COLORING AND STYLE

Enter a **Point radius**. Select the **Radius scale factor** check box to enter a number to scale the value more than the default of 1. Select a **Color** for the point.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. To prevent a style attribute from being inherited, click to clear the **Color**, **Color and data range**, or **Point radius scale factor** check boxes, or select **None** from the **Plot** list.

Point Graphs

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 these to a **1D Plot Group** or **Polar Plot Group**.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**—**From parent**, **Solution**, or **Cut Point**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. If a **Parametric Solver** or **Parametric Sweep** data set is selected, the **Parameter values** section lists the associated parameter values. Select an option from the **Select via** list—**Stored output times** or **Interpolated times**. If **Interpolated times** is selected, also enter **Times**.

Y-AXIS DATA OR R-AXIS DATA

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

X-AXIS DATA OR θ ANGLE DATA (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. 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.

Tip

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.

From the **Parameter** list, select—**Parameter value**, **Time**, **Solution number**, **Frequency spectrum** or **Expression**.

- If **Frequency spectrum** is selected, COMSOL Multiphysics computes the number of frequencies and the frequency range based on the FFT 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 or select the **Frequency range** check box and then enter the range in the **Minimum** and **Maximum** fields (SI unit: Hz).
- If **Expression** is selected, click the **Replace Expression** (⇄ ▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select a **Unit** and select the check box to enter a **Description** (or edit the default).

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.

COLORING AND STYLE

In the **Line style** section, select a **Line** type—**None**, **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. Select a **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Enter a line **Width** (the default is 1) or use the slider to select.

In the **Line markers** section, select a **Marker**—**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) 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).

LEGENDS

If required, select the **Show legends** check box to display text to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, these values also display. For point plots, the legend displays the coordinate (or vertex number). If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

Principal Stress Volume Plots

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, orders 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 shows the corresponding eigenvectors using arrows. These plots are found on the context menu under **More Plots**.

Right-click to add **Deformation**, **Filter**, or **Color Expression** subnodes to deform or color the principal stress plot's arrows.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

PRINCIPAL COMPONENTS

Select a **Type**—**Principal stress** or **Principal strain.** Under **Principal values**, enter information for the **First Value**, **Second Value**, 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. The defaults are the directions (eigenvectors) for the first, second, and third principal stresses. For transient problems, enter a **Time**.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

POSITIONING

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). If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

COLORING AND STYLE

Select an **Arrow type**—**Arrow** or **Cone**. Select an **Arrow length**—**Proportional** (the default), **Normalized**, or **Logarithmic**. If **Logarithmic** is selected, also enter a value for the **Range quotient**. Select the position of the **Arrow base**—**Tail** (the default) or **Head**. Enter a **Scale factor** or use the slider to select a value.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Principal Stress Surface Plots

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, orders 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 shows the corresponding eigenvectors using arrows. These plots are found on the context menu under **More Plots**.

Right-click to add **Deformation**, **Filter**, or **Color Expression** subnodes to deform or color the principal stress plot's arrows.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

PRINCIPAL COMPONENTS

Select a **Type**—**Principal stress** or **Principal strain.** Under **Principal values**, enter information for 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. The defaults are the directions (eigenvectors) for the first, second, and third principal stresses. For transient problems, enter a **Time**.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

POSITIONING

2D

For **2D Principal Stress Surface Plots** ( ), select an **Entry method**—**Number of points** or **Coordinates** for the **x grid points** and **y grid points**. If **Number of points** is selected, enter the number of **Points** in each direction. (The default is 15). If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

COLORING AND STYLE

Select an **Arrow type**—**Arrow** or **Cone**. Select an **Arrow length**—**Proportional** (the default), **Normalized**, or **Logarithmic**. If **Logarithmic** is selected, also enter a value for the **Range quotient**. Select the position of the **Arrow base**—**Tail** (the default) or **Head**. Enter a **Scale factor** or use the slider to select a value. Select a **Placement**—**Uniform** or **Elements**.

3D

For **3D Principal Stress Surface Plots** ( ), and if **Uniform** is selected as the **Placement**, also enter the total **Number of arrows** (default is 200).

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Arrow scale factor**, **Color**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Scatter Surface and Scatter Volume Plots

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). These plots are found on the **More Plots** submenu.

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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Under **Expression,** enter or select **x component** and **y component** (2D), **r component** and **z component** (2D axial symmetry), or **x component**, **y component**, and **z component** (3D). Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. The expressions become the quantities that span the axes in the rectangular or block grid where the spheres (disks) of the scatter plots appear. Enter or edit a **Description**. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

EVALUATION POINTS

Under **Evaluation Points**, select an **Entry method** for the **x grid points** and **y grid points** (2D), **r grid points** and **z grid points** (2D axial symmetry), or for the **x grid points**, **y grid points**, and **z grid points** (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 Principal Stress Surface and 7 for 3D Principal Stress Volume).
- If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

RADIUS

Under **Radius**, the default **Expression** is 1 (SI unit: m). Enter any **Expression** in the field. COMSOL automatically scales the radius. Select the **Radius scale factor** check box to edit the default of 1 and enter a scalar number.

COLOR

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**. If the default **Color table** (**Rainbow**) is not suitable for the plot, try other options. If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color** and **Point radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Slice Plots

Use a **Slice Plot** () to display a scalar quantity on slices inside a 3D domain. Add **Deformation** and **Filter** subnodes as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

PLANE DATA

Specify the slices from the **Plane Type** list—**Quick** (default), or **General**.

- If **Quick** is selected, from the **Plane** list, select **xy-planes**, **yz-planes**, or **zx-planes**. 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** or click the **Range** button ( ) to define a range of values.
- 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) or click the **Range** button ( ) to define a range of values.

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.

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.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Select a **Color table**. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**.

The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to control where the recovery operation applies—**Within domains** or **Everywhere**.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Streamline Plots

A **Streamline** plot in 2D ( ) or 3D ( ) visualizes 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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

Note

Go to Streamline Plots in the *COMSOL Multiphysics User's Guide* for detailed instructions.

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter or select values for the **x component** and **y component** (2D), or the **x component**, **y component**, and **z component** (3D). Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined quantities based on the physics of the model.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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 in the *COMSOL Multiphysics User's Guide*:

- 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

Note

The **Selection** section is made available for some data sets when **On selected boundaries** is selected 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.

COLORING AND STYLE

Select a **Line type**—**Line**, **Ribbon** (3D only), or **Tube**.

- If **Ribbon** is selected, enter a width (SI unit: m) for the ribbons in the **Width expression** field. 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.
- If **Tube** is selected, enter a **Tube radius expression** (the radius of the tube; SI unit: m). To edit the **Radius scale factor** select the check box and enter a number.

For all line types, unless you use **Color Expression** subnode for the streamline colors, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

Ribbons are an alternative to tubes for visualization of, for example, the vorticity of a flow field.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field. The **Recover** default is **Off** because accurate derivative recovery takes processing time. If required, select another option from the **Recover** list—**Within domains** or **Everywhere**.

ADVANCED

Edit the **Integration tolerance**, **Maximum number of integration steps**, the **Maximum integration time**, the **Stationary point stop tolerance**, and the **Loop tolerance**. 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.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, **Deform scale factor**, and **Tube radius scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Surface Plots

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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** (▶ ▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**.

- If **Color table** is selected, choose a **Color table** from the list (default: Rainbow). The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.
- If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

To plot only on the visualization mesh, select the **Wireframe** check box. This displays the surface as a triangular grid.

QUALITY

! *Important*

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field.

Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**. Choose **Internal** to prevent smoothing across material boundaries, for example. The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to specify where the recovery applies—**Within domains** or **Everywhere**.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, **Deform scale factor**, and **Height scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Table Graph

Add a **Table Graph** ( ) plot to a **1D Plot Group** or **Polar Plot Group** to display data from a table with one line per output column.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Table** or click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. In the **x-axis data** list (or θ **angle data** for polar plots), select the column to use 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.

COLORING AND STYLE

Select a **Line style** from the **Line** list—**None**, **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. Select a **Color**—any basic color, **Cycle**, which cycles among these colors, or **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Enter a line **Width** or use the slider to select. In the **Line markers** section, select a **Marker**—**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) 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).

LEGENDS

The **Show legends** check box is selected by default to display to the right of the plot. If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

Table Surface Plots

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). This plot is available from the **More Plots** submenu and also by selecting **Surface** ( ) from the **Results** window's toolbar.

Note

When created from the **Results** window's 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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Table**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. 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.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**.

- If **Color table** is selected, choose a **Color table** from the list (default: Rainbow). The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.
- If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

To plot only on the visualization mesh, select the **Wireframe** check box. This displays the surface as a triangular grid.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, **Deform scale factor**, and **Height scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Volume Plots

Use a **Volume** () plot to display a quantity inside a domain in 3D. Add **Deformation** or **Filter** subnodes as required.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

DATA

Select a **Data set**. **From parent** 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 node that the selection in the list next to the button refers to.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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.

COLORING AND STYLE

Select a **Coloring**—**Color table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens. Select a **Color table**. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

To plot only on the visualization mesh, select the **Wireframe** check box. This displays the surface as a triangular grid.

QUALITY

Important

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

Select a plot **Resolution**—**Extra fine, Finer, Fine**, **Normal**, **Coarse**, **No refinement** (the default), or **Custom**. Select **Custom** to manually define the number of refinements of each mesh element in the **Element refinement** field.

Select an option from the **Smoothing** list—**None**, **Internal**, or **Everywhere**. The **Recover** default is **Off** because accurate derivative recovery takes processing time. To use accurate derivative recovery, select another option from the **Recover** list to specify where the recovery applies—**Within domains** or **Everywhere**.

ELEMENT FILTER

Select the **Enable filter** check box and select a **Criterion**—**Random**, **Expression**, or **Logical expression**. If **Expression** or **Logical expression** is selected, enter an **Expression** in the field. If **Random** or **Expression** is selected, specify the **Fraction** of elements to show, or use the slider to select a value.

SHRINK ELEMENTS

Enter an **Element scale factor** between 0 and 1 to scale elements in the mesh plot.

INHERIT STYLE

Use this feature to maintain a consistent style between plots. After there is more than one plot in a plot group, select a **Plot** from the list to inherit the style attributes. The **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Color Expressions

Use a **Color Expression** feature ( ) to add coloring to the shapes defined by a plot. Add this to these 2D and 3D plots—arrow, contour, isosurface, particle tracing, and streamline.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

RANGE

Manually override the **Minimum** and **Maximum color range** or **data range**. Click to select the check box, then specify a range using the fields or use the sliders to control values.

COLORING AND STYLE

Select an appropriate **Color table** for the plot based on your audience and what you plan to do with the final analysis. The **Color legend** is selected by default and displays to the right of the plot. Select the **Reverse color table** check box to reverse the order of the colors in the color table. Select the **Symmetrize color range** check box to obtain a color range centered around zero.

Deformations

Add a **Deformation** feature ( ) to deform the plot according to a vector quantity, for example, the displacement field in structural mechanics. Add this 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.

Tip	Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

EXPRESSION

Enter an **Expression** and enter or edit a **Description**. Enter **x**-, **y-**, and **z-components** as required for the space dimension. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

SCALE

Select the **Scale factor** check box to edit the default value for the scale factor.

Filters

Add a **Filter** ( ) to the following 2D and 3D plots—arrow, contour, isosurface, line plot, slice, and volume. 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.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

ELEMENT SELECTION

Enter a **Logical expression for inclusion**; for example, type `x>0.5` to include only the part of the plot that is located in a region where the *x* coordinate is greater than 0.5. Select the **Element nodes to fulfill expression**—**All** (the default), **At least one**, or **At least one but not all**. This determines if the filtered plot contains elements where all their nodes, at least one node, or at least one but not all nodes fulfill the criterion in the logical expression.

Height Expressions

The **Height Expression** subnode ( ) introduces 3D height on a 2D surface plot or 2D table surface plot. Add it to make the height of the plot represent a scalar quantity. 2D Surface, 2D Line, and 2D Table Surface plots support the height expression attribute.

Tip

Links to all the other plots are found in Table 6-2 (1D and polar) and Table 6-3 (2D and 3D).

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, enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

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).

Derived Values and Tables

In this section:

- Derived Values
- Volume Average, Surface Average, and Line Average Derived Values
- Volume Integration, Surface Integration, and Line Integration Derived Values
- Volume, Surface, and Line Maximums and Minimums for Derived Values
- Point Evaluation Derived Values
- Global Evaluation Derived Values
- Global Matrix Evaluation
- Tables (Derived Values)

Derived Values

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** to define evaluations of numerical results—globally, in a point, or integrated quantities. For 2D and 3D plots, also get numerical results directly in a table by clicking the plot.

For all derived values 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, 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 provides 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.

TABLE 6-4: DERIVED VALUE TYPES

DERIVED VALUE	ICON	DESCRIPTION	LINK TO SECTION
Point Evaluation	8.85 e-12	To evaluate expressions or variables defined in a point.	Point Evaluation Derived Values
Global Evaluation	8.85 e-12	To evaluate the numerical value of a global variable.	Global Evaluation Derived Values

TABLE 6-4: DERIVED VALUE TYPES

DERIVED VALUE	ICON	DESCRIPTION	LINK TO SECTION
AVERAGE SUBMENU			
Volume Average		To evaluate an average over a set of domains in 3D models.	Volume Average, Surface Average, and Line Average Derived Values
Surface Average		To evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
LINE AVERAGE		To evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.	
INTEGRATION SUBMENU			
Volume Integration	∭	To evaluate an integral over a set of domains in 3D models.	Volume Integration, Surface Integration, and Line Integration Derived Values
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		To evaluate the maximum over a set of domains in 3D models.	Volume, Surface, and Line Maximums and Minimums for Derived Values
Surface Maximum		To evaluate the maximum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
Line Maximum		To evaluate the maximum over a set of domains in 1D, boundaries in 2D, or edges in 3D.	

TABLE 6-4: DERIVED VALUE TYPES

DERIVED VALUE	ICON	DESCRIPTION	LINK TO SECTION
MINIMUM SUBMENU			
Volume Minimum	∿	To evaluate the minimum over a set of domains in 3D models.	Volume, Surface, and Line Maximums and Minimums for Derived Values
Surface Minimum	∿	To evaluate the minimum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
Line Minimum	∿	To evaluate the minimum over a set of domains in 1D, boundaries in 2D, or edges in 3D.	

ADDING A DERIVED VALUE NODE

1 In the **Model Builder**, under **Results**, right-click **Derived Values** (8.85 e-12).

2 Select an option from the list and continue defining each derived value (see Table 6-4).

3 After adding a **Derived Value** node, click the **Evaluate** button (≡) or right-click the **Derived Values** node and select **Evaluate All** (≡) or **Clear and Evaluate All** (≡).

4 **Evaluate All** calculates and displays the results in the derived values table for all nodes in the Model Builder sequence and keeps the previous data. **Clear and Evaluate All** resets and clears all the data and calculates from the beginning.

See Also

- Tables (Derived Values)

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- Defining Derived Values and Tables
- Derived Values Common Settings
- Expressions and Predefined Quantities
- Editing and Organizing Results Tables
- Getting Numerical Results Directly
- Entering Ranges and Vector-Valued Expressions
- Solver Studies and Study Types
- Working with Geometry

Volume Average, Surface Average, and Line Average Derived Values

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. Select:

- **Volume Average** ( ) to evaluate an average over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Results** window.
- **Surface Average** ( ) to evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Average** ( ) to evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.

DATA

Select a **Data set**. The result is computed by averaging over the data set selected. Only data sets with data for an applicable space dimension are available. For **Parametric Solver** and **Parametric Sweep** studies, the **Parameter values** section lists the associated parameter values.

Parametric Sweep Studies

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 on the rows in the table.

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.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

INTEGRATION SETTINGS

Select a **Method**. The **Auto** settings computes the integral for fields by numerical integration and for reaction forces by summation. Override by choosing **Integration** or **Summation**.

DATA SERIES OPERATION

In addition to performing, for example, 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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the *root mean square* or *quadratic mean*), **Standard deviation**, or **Variance**. If **Maximum** is selected, select an option from the **Find maximum of** list—**Real part** or **Absolute value**. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.

DATA SERIES OPERATION

In addition to performing, for example, 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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the *root mean square* or *quadratic mean*), **Standard deviation**, or **Variance**.

Volume Integration, Surface Integration, and Line Integration Derived Values

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. 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 **Results** 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.

Model

- See Effective Diffusivity in Porous Materials for a surface integration example, Model Library path **COMSOL_Multiphysics>Diffusion>effective_diffusivity**.
- If you have the AC/DC Module, see An RFID System for a line integration example, Model Library path **ACDC_Module>General_Industrial_Applications>rfid**.

DATA

Select a **Data set**. The result is computed by integrating over the data set selected. Only data sets with data for an applicable space dimension are available. 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.

Parametric Sweep Studies

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 on the rows in the table.

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.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** (⇆ ▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

INTEGRATION SETTINGS

Select a **Method**.

The **Auto** settings computes the integral for fields by numerical integration and for predefined reaction forces by summation. Also, if an expression is specified for the integrand using the `reacf` operator, the automatic setting chooses the summation method.

Override by choosing **Integration** or **Summation**:

- **Integration** is the standard numerical integration method (quadrature).
- **Summation** is a summation method that 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.

For the integration order, the **Auto** check box is selected by default. The program then determines an appropriate integration order for the expression. Clear the **Auto** check box to make an entry in the **Integration order** field. The program then performs the integration elementwise using a quadrature rule of the selected order.

DATA SERIES OPERATION

In addition to performing, for example, 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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the root mean square or quadratic mean), **Standard deviation**, or **Variance**. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.

Volume, Surface, and Line Maximums and Minimums for Derived Values

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.

- Use **Volume Maximum** (∿) or **Volume Minimum** (∿) 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 **Results** window.

- Use **Surface Maximum** ( ) or **Surface Minimum** ( ) to evaluate a maximum or minimum value over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- Use **Line Maximum** ( ) or **Line Minimum** ( ) to evaluate a maximum or minimum value over a set of domains in 1D, boundaries in 2D, or edges in 3D.

DATA

Select a **Data set**. The result is computed by integrating over the data set selected. Only data sets with data for an applicable space dimension are available.

Parametric Sweep Studies

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 on the rows in the table.

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.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

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.

DATA SERIES OPERATION

In addition to performing, for example, 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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the root mean square or quadratic mean), **Standard deviation**, or **Variance**.

Point Evaluation Derived Values

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** and displayed in the **Results** window.

Model

If you have the Acoustics Module, see Eigenmodes in a Muffler: Model Library path **Acoustics_Module>Industrial_Models>muffler_eigenmodes**.

DATA

Select a **Data set**.

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.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

DATA SERIES OPERATION

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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the root mean square or quadratic mean), **Standard Deviation**, or **Variance**. If **Maximum** is selected, select an option from the **Find maximum of** list—**Real part** or **Absolute value**. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.

Global Evaluation Derived Values

Use a **Global Evaluation** (8.85 e-12) to define the evaluation of the numerical value of a global variable.

DATA

Select a **Data set**. The result is computed by integrating over the data set selected.

EXPRESSION

Enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** (▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

DATA SERIES OPERATION

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. Select an **Operation**—**None** (the default), **Average**, **Integral**, **Maximum**, **Minimum**, **RMS** (the root mean square or quadratic mean), **Standard Deviation**, or **Variance**. If **Maximum** is selected, select an option from the **Find maximum of** list—**Real part** or **Absolute value**. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.

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.

DATA

Select a **Data set**. The result is computed for all solution data in the selected data set. The data set typically contains solutions from a parametric sweep

EXPRESSION

Select a predefined matrix variable using the **Replace Expression** (▾) button. Select a **Unit**, and enter or edit a **Description**. Select the **Description** check box to edit the default description for a predefined variable.

DATA SERIES OPERATION

Select from the following options for operations on the data series for the inner solutions—typcially 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** (not a number) 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.

Tables (Derived Values)

Tables can store the result of **Derived Values** and results from probes, for example. The result are displayed in the **Results** window, which by default is located below the **Graphics** window. There are two ways to add a **Table**:

- In the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, click the **Evaluate (New Table)** button ( ). This evaluates the feature and stores the result in a new table. This button is also used as the **Evaluate** button.
- In the **Model Builder**, right-click the **Tables** node, and select **Table**.

  A **Table** node is added under **Tables**. Click the node to display a table with the selected integration feature's description and values in the **Results** 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's **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.

DATA

If applicable (for accumulated probe tables, for example) select a format from the **Format** list:

- Select **All** (the default) to display 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.
  - 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.

Note

The imported data replaces any existing data in the table.

Exports

In this section:

- Export Feature Types
- Animation Export
- Data Export
- Mesh Export
- Plot 1D, 2D, or 3D Image Export
- Plot Export
- Player

Export Feature 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 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 6-5.

TABLE 6-5: EXPORT TYPES

EXPORT TYPE	ICON	DESCRIPTION
Animation Export	🎞	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 Export	➡	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.
Plot 1D, 2D, or 3D Image Export	🖼	To export plot images from 1D, 2D, and 3D Plot Groups
Player	▶	To create interactive animations.
Plot Export	➡	To export a plot from a 1D, 2D, or 3D Plot Group.

See Also

In the *COMSOL Multiphysics User's Guide*:

- Results Evaluation and Visualization
- Exporting Data and Images

Animation Export

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.

Important

Use Windows Media Player to play AVI files.

Tip

Use this feature, for example, to export multiple images for different time steps or eigenvalues.

SCENE

Select a **Subject**, which is one of the 1D, 2D, or 3D plot groups previously defined.

OUTPUT

Select an **Output type**—**Image sequence** generates multiple image files; **Movie** generates a single file containing all the images. Enter or **Browse** a **Filename**, select a **Format** for movies—**GIF**, **Flash**, or **AVI**. Select the number of **Frames per second**. For **Flash** and **GIF**, select the **Open in browser** check box to launch the default web browser to view the output file. For **Flash**, select or clear the **Interpolate between frames** check box. For **AVI**, specify the **Quality** as a scalar number between 0 and 1 (default: 0.75).

Important

The AVI format is not supported on Windows XP. For movies using the AVI format, you might also need to change the codec used for animations (in the **Graphics** section of the **Preferences** dialog box).

FRAME SETTINGS

From the **Size** list, select **Manual** (the default) or **Current**, which uses the current size of the **Graphics** window. If you select **Current**, the **Record in reverse order** check box is the only additional setting. Keep the original animation width and height with the **Lock aspect ratio** check box or enter the number of pixels (px) for the generated image size in the **Width** and **Height** fields (default value: 640-by-480 pixels). Select the **Record in reverse order** check box if required.

PARAMETER SWEEP

Select a **Sweep type**—**Result parameter**, **Dynamic data extension**, or **Stored solutions** (the default). Define the **Sweep type** parameters as required.

Layout

By default, the **Title**, **Legend**, **Axes**, and **Logotype** are included. To edit the default, select the **Include** check box and click to clear one or several of the **Title**, **Legend**, **Axes**, and **Logotype** check boxes.

Enter a **Font size** (pt). Select a **Background** from the list: **Current** or **Color**. If you select **Color** (the default setting), click the **Color** button 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

This section is by default not expanded. Enter a **Resolution** for the images in the animation. The default is 96 DPI (dots per inch). Select the **Antialiasing** check box to reduce stairstep-like lines and to smooth lines and edges.

Data Export

Use **Data** ( ) to export numerical data to file. Data export operates directly on data sets. You can use different types of evaluation points other than the data set (for example, a grid) and you can export the data on spreadsheet format or a sectionwise format.

DATA

Select a **Data set**. The **Data set** list contains the 1D, 2D, or 3D data sets previously defined, for example, **Solution** or **Cut Line**.

EXPRESSIONS

Specify the **Expression** to output. Click the **Replace Expression** ( ) or **Add Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression** in the table or field.

Enter a **Description** in the table or field. Use the **Move Up** ( ), **Move Down** ( ), and **Delete** ( ) buttons as required to move rows in the table.

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.

From the **Points to evaluate in** list, select the source for the points to evaluate the data in:

- If you select **From file**, specify a **Coordinate filename** for a text file with the coordinates for the data output, or click **Browse** to locate the file.
- If you select **Grid**, then from the **Data format** list, select **Sectionwise** or **Spreadsheet**. Specify the **x**, **y**, and **z** coordinates for the grid points in the fields or using the **Range** buttons ( ).
- If you select **Regular grid**, then from the **Data format** list, select **Sectionwise** or **Spreadsheet**. Specify the **Number of x points**, **Number of y points**, and **Number of z points** for the regular grid in the fields (default: 10 points in each direction)
- If you select **Take from data set** (the default), then from the **Data format** list, select **Sectionwise** or **Spreadsheet**. From the **Space dimension** list, select **Take from data set** or **Manual**. If **Manual** is selected, enter a **Dimension**. From the **Geometry level** list, select **Take from data set**, **Volume**, **Surface**, **Line**, or **Point**.

ADVANCED

The **Include header** and **Full precision** check boxes are selected by default. Click to clear these check boxes to remove the headers and limit the precision in the output to seven digits, respectively.

Select the **Sort** check box to sort the data by increasing x, y, and z coordinates. Otherwise, the data is unsorted. Select **Lagrange points** (the default) or **Gauss points** from the **Evaluate in** list to specify where COMSOL evaluates the data: in the Lagrange points (nodes of the Lagrange elements) or in the Gauss points for the Gaussian quadrature.

Select a **Resolution**—**Finer**, **Fine**, **Normal** (default), or **Custom**. If **Custom** is selected, enter a **Lagrange-element node-point order** (default: 1). Use a higher node-point order for a finer resolution.

Select a **Smoothing** method for smoothing of the data—**None**, **Internal** (the default, for smoothing within domains but not across interior boundaries), or **Everywhere**.

Mesh Export

Use the **Mesh** ( ) feature 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.

You can also export then mesh data 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**.

DATA

Select a **Data set**. The **Data set** list contains the solution and mesh data sets previously defined.

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. You can save the mesh using two formats: binary `.mphbin` files or as text files with the extension `.mphtxt`.

Plot 1D, 2D, or 3D Image Export

Use the **1D Image**, **2D Image**, and **3D Image** ( ) features to export plot images from 1D, 2D, and 3D plot groups respectively.

Tip

Use the Animation Export feature, for example, to export multiple images for different time steps or eigenvalues.

SCENE

Select a **Subject** from the list. The **Subject** 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.

IMAGE

2D

3D

For 2D and 3D models, select a **View**—**From plot group** uses the plot group settings, or select a predefined **View** setting from the list.

On the toolbar, click the **Refresh** button (⟳) to refresh the **Graphics** window with your settings. This is useful when you change the Plot group or View selection.

From the **Size** list, select **Manual** to specify the image size manually using the settings below, or select **Current** to use the current size of the **Graphics** windows (in that case, only the **Antialiasing** setting is available). Select a **Unit** of dimension—**Pixels (px)**, **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).

Select the **Antialiasing** check box (selected by default) to reduce stairstep-like lines and to smooth lines and edges.

FILE

Choose a format for the image file from the **Format** list: PNG, BMP, JPEG, or (1D only) EPS. The default format is PNG.

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.

LAYOUT

By default, the **Title**, **Legend**, **Axes**, and **Logotype** are included. To edit the default, select the **Include** check box and click to clear one or several of the **Title**, **Legend**, **Axes**, and **Logotype** check boxes. You can also enter a **Font size** (pt). Select a **Background** from the list: **Current** or **Color**. If you select **Color** (the default setting), click the **Color** button 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.

Plot Export

Use **Plot** () to export a plot from a 1D, 2D, or 3D Plot Group.

PLOT

Select a **Plot group** from the list The **Plot group** list contains the **1D**, **2D**, or **3D Plot Groups** previously defined. 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 **Sectionwise** or **Spreadsheet**. 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. 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.

ADVANCED

The **Include header** and **Full precision** check boxes are selected by default. Click to clear the check box to remove the headers and to limit the precision in the output to seven digits, respectively. Select the **Sort** check box to sort the data by increasing x, y, and z coordinates. Otherwise, the data is unsorted.

Player

A **Player** () creates interactive animations directly in the COMSOL Desktop.

SCENE

Select a **Subject** for the player. The **Subject** list contains the 1D, 2D, and 3D plot groups previously defined and **None**.

PARAMETER SWEEP

Select a **Sweep type**—**Result parameter**, **Dynamic data extension**, or **Stored solutions** (the default).

FRAMES

If the selected sweep type is **Stored solutions**, select **All** or **Number of frames** from the **Frame selection** list. Select **All** to play all solutions in the stored solution. Also enter the

Number of frames (if applicable). The default is 25. 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

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.

Right-click the **Player** node and select **Generate Frame** (🎞) 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 these buttons on the **Graphics** window to **Play** (▶), **Stop** (■), **Next** (⏭), and **Previous** (⏮), to cycle through the animation.

Reports

In this section:

- Report Feature Types
- Generating a Model Report

Report Feature Types

For any model you can add one or more reports to the **Reports** branch and then generate model reports in HTML format for documentation and information about a model. You can create reports using four predefined templates that define different levels of detail in the report:

- Complete Report: Includes all information about a model.
- Intermediate Report: Includes most information about the geometry, mesh, physics, studies, and results, including image snapshots that shows the location of boundary conditions and other physics settings.
- Brief Report: Includes the most important information about the geometry, mesh, physics, studies, and results. It does not include images showing where all physics settings are active, for example.
- Custom Report: This report is initially empty. You can add report nodes freely to create a custom report.

Note

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 that you add afterward.

Generating a Model Report

For a model for which you want to create are report, the following steps describe the general procedure:

1. Right-click **Report** ( ) and choose a template that creates report nodes that describe the model with a suitable level of detail: **Custom Report**, **Brief Report**, **Intermediate Report**, or **Complete Report**.

2. In the main **Report** node' **Settings** window, specify the location for the HTML file and the associated folder with images and stylesheet information.
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.

7

The Finite Element Method

This chapter contains a theoretical background to the finite element method and an overview of the finite element types in COMSOL Multiphysics. Sections also explain how COMSOL Multiphysics forms the system of equations and constraints that it solves and the implications of Dirichlet conditions involving several solution components in a multiphysics model. The chapter also includes a theoretical background to the weak formulation of PDEs and an overview of PDE, energy, and weak form formulations.

The Finite Element Method

This section describes how the Finite Element Method (FEM) approximates a PDE problem with a problem that has a finite number of unknown parameters, that is, a *discretization* of the original problem. This concept introduces *finite elements*, or *shape functions*, that describe the possible forms of the approximate solution.

In this section:

- Mesh
- Finite Elements
- Discretization of the Equations

Mesh

The starting point for the finite element method is a mesh, a partition of the geometry into small units of a simple shape, *mesh elements*.

Sometimes the term “mesh element” means any of the mesh elements—mesh faces, mesh edges, or mesh vertices. When considering a particular d-dimensional geometric entity in the geometry (that is, a domain, boundary, edge, or vertex), then by its mesh elements you mean the d-dimensional mesh elements contained in that geometric entity.

Note

For more information about the types of elements available in 1D, 2D, and 3D, see About Mesh Elements for 1D, 2D, and 3D Geometries in the *COMSOL Multiphysics User’s Guide*.

Finite Elements

Once you have a mesh, you can introduce approximations to the dependent variables. For this discussion, concentrate on the case of a single variable, u. The idea is to approximate u with a function that you can describe with a finite number of parameters, the so-called *degrees of freedom* (DOF). Inserting this approximation into the weak form of the equation generates a system of equations for the degrees of freedom.

Start with a simple example: linear elements in 1D. Assume that a mesh consists of just two mesh intervals: $0 < x < 1$ and $1 < x < 2$. Linear elements means that on each mesh interval the continuous function u is linear (affine). Thus, the only thing you need to know in order to characterize u uniquely is its values at the *node points* $x_1 = 0$, $x_2 = 1$, and $x_3 = 2$. Denote these as $U_1 = u(0)$, $U_2 = u(1)$, $U_3 = u(2)$. These are the *degrees of freedom*.

Now you can write

$$u(x) = U_1\varphi_1(x) + U_2\varphi_2(x) + U_3\varphi_3(x)$$

where $\varphi_i(x)$ are certain piecewise linear functions. Namely, $\varphi_i(x)$ is the function that is linear on each mesh interval, equals 1 at the i^{th} node point, and equals 0 at the other node points. For example,

$$\varphi_1(x) = \begin{cases} 1 - x & \text{if } 0 \le x \le 1 \\ 0 & \text{if } 1 \le x \le 2 \end{cases}$$

The $\varphi_i(x)$ are called the *basis functions.* The set of functions $u(x)$ is a linear function space called the *finite element space.*

For better accuracy, consider another finite element space corresponding to quadratic elements. Functions u in this space are second-order polynomials on each mesh interval. To characterize such a function, introduce new node points at the midpoint of each mesh interval: $x_4 = 0.5$ and $x_5 = 1.5$. You must also introduce the corresponding degrees of freedom $U_i = u(x_i)$. Then, on each mesh interval, the second-degree polynomial $u(x)$ is determined by the degrees of freedom at the endpoints and the midpoint. In fact, you get

$$u(x) = U_1\varphi_1(x) + U_2\varphi_2(x) + U_3\varphi_3(x) + U_4\varphi_4(x) + U_5\varphi_5(x)$$

where the basis functions $\varphi_i(x)$ now have a different meaning. Specifically, $\varphi_i(x)$ is the function that is quadratic on each mesh interval, equals 1 at the ith node point, and equals 0 at the other node points. For example,

$$\varphi_1(x) = \begin{cases} (1 - x)(1 - 2x) & \text{if } 0 \le x \le 1 \\ 0 & \text{if } 1 \le x \le 2 \end{cases}$$

In general, you specify a finite element space by giving a set of basis functions. The description of the basis functions is simplified by the introduction of *local coordinates* (or *element coordinates*). Consider a mesh element of dimension d in an

n-dimensional geometry (whose space coordinates are denoted $x_1,\ldots, x_n$). Consider also the *standard d-dimensional simplex*

$$\xi_1 \geq 0, \xi_2 \geq 0, \ldots, \xi_d \geq 0, \xi_1 + \ldots + \xi_d \leq 1$$

which resides in the local coordinate space parameterized by the local coordinates ξ_1, …, ξ_d. If $d = 1$, then this simplex is the unit interval. If $d = 2$, it is a triangle with two 45 degree angles, and if $d = 3$ it is a tetrahedron. Now you can consider the mesh element as a linear transformation of the standard simplex. Namely, by letting the global space coordinates x_i be suitable linear (affine) functions of the local coordinates, you get the mesh element as the image of the standard simplex.

When described in terms of local coordinates, the basis functions assume one of a few basic shapes. These are the *shape functions*. In the example with linear elements in 1D, any basis function on any mesh element is one of the following:

$$\phi = \xi_1, \qquad \phi = 1 - \xi_1, \qquad \phi = 0$$

Thus the first two are the shape functions in this example (0 is not counted as a shape function). In the example with quadratic elements in 1D, the shape functions are

$$\phi = (1-\xi_1)(1-2\xi_1), \qquad \phi = 4\xi_1(1-\xi_1), \qquad \phi = \xi_1(2\xi_1 - 1)$$

CURVED MESH ELEMENTS

When using higher-order elements (that is, elements of an order > 1), the solution has a smaller error. The error also depends on how well the mesh approximates the true boundary. To keep errors in the finite element approximation and the boundary approximation at the same level, it is wise to use *curved mesh elements*. They are distorted mesh elements that can approximate a boundary better than ordinary straight elements (if the model's boundary is curved). You can get curved mesh elements by writing the global coordinates x_i as polynomials of order k (the *geometry shape order*) in the local coordinates ξ_j (the earlier example took $k = 1$). Then the mesh element is the image of the standard simplex. For mesh elements that do not touch the boundary, there is no reason to make them curved, so they are straight. It is customary to use the same order k here as for the order of the (Lagrange) element. This is referred to as using *isoparametric elements*.

The order k is determined by the geometry shape order for the frame (coordinate system) associated with the finite element. The frame is determined by the property `frame` to the finite element (the default is the reference frame). For certain finite elements, the geometry shape order given by the frame can be overridden by the property `sorder`. In the COMSOL Multiphysics user interface, the default setting is to use an automatic geometry shape order, which means that the geometry shape order is equal to the highest order of any shape function used in the model.

If a curved mesh element becomes too distorted, it can become inverted and cause problems in the solution. The software can then reduce the geometry shape order automatically to avoid inverted elements.

See Also

In the *COMSOL Multiphysics User's Guide:*

- Shape Function Variables
- Avoiding Inverted Mesh Elements

THE LAGRANGE ELEMENT

The preceding examples are special cases of the *Lagrange element*. Consider a positive integer k, the *order* of the Lagrange element. The functions u in this finite element space are piecewise polynomials of degree k, that is, on each mesh element u is a polynomial of degree k. To describe such a function it suffices to give its values in the *Lagrange points* of order k. These are the points whose local (element) coordinates are integer multiples of $1/k$. For example, for a triangular mesh in 2D with $k = 2$, this means that you have node points at the corners and side midpoints of all mesh triangles. For each of these node points p_i, there exists a degree of freedom $U_i = u(p_i)$ and a basis function φ_i . The restriction of the basis function φ_i to a mesh element is a polynomial of degree (at most) k in the local coordinates such that $\varphi_i = 1$ at node i, and $\varphi_i = 0$ at all other nodes. Thus the basis functions are continuous and you have

$$u = \sum_i U_i \varphi_i$$

The Lagrange element of order 1 is called the linear element. The Lagrange element of order 2 is called the quadratic element.

The Lagrange elements are available with all types of mesh elements, and it is the most common element type in the COMSOL Multiphysics physics interfaces. The order k can be arbitrary, but the available numerical integration formulas usually limits its usefulness to $k \leq 5(k \leq 4$ for tetrahedral meshes).

Syntax for the Lagrange Element (shlag)

Specify Lagrange elements in the `model.shape` field of the model object. The constructor of the Lagrange shape function is `shlag`. The following properties are allowed:

TABLE 7-1: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHLAG SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name
`order`	Positive integer		Basis function order
`border`	Positive integer		Alias for order
`sorder`	Positive integer	Determined by frame	Geometry shape order

It is not possible to abbreviate the property names, and you must write them in lowercase letters enclosed in quotation marks. For example:

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shlag");
model.shape("shu").feature("f1").set("order",2);
model.shape("shu").feature("f1").set("basename","u");
```

The Lagrange element defines the following variables. Denote `basename` with u, and let x and y denote (not necessarily distinct) space coordinates. The variables are (sdim = space dimension and edim = mesh element dimension):

- u
- ux, meaning the derivative of u with respect to x, defined on edim = sdim
- uxy, meaning a second derivative, defined on edim = sdim
- uTx, the tangential derivative variable, meaning the x-component of the tangential projection of the gradient, defined on edim < sdim
- $uTxy$, meaning xy-component of the tangential projection of the second derivative, defined when edim < sdim

When calculating the derivatives, the global space coordinates are expressed as polynomials of degree (at most) `sorder` in the local coordinates.

Note

The use of isoparametric elements means that u is not a polynomial in the global coordinates (if $k > 1$), only in the local coordinates.

THE ARGYRIS ELEMENT

For a function represented with Lagrange elements, the first derivatives between mesh elements can be discontinuous. In certain equations (for example, the biharmonic equation) this can be a problem. The *Argyris element* has basis functions with continuous derivatives between mesh triangles (it is defined in 2D only). The second order derivative is continuous in the triangle corners. On each triangle, a function *u* in the Argyris finite element space is a polynomial of degree 5 in the local coordinates.

The Argyris element is available with triangular meshes only.

When setting Dirichlet boundary conditions on a variable that has Argyris shape functions, a locking effect can occur if the boundary is curved and constraint order (`cporder`) `5` is used. Use `cporder=4` if the boundary is curved and `cporder=5` for straight boundaries.

Syntax for the Argyris Element (sharg_2_5)

Specify Argyris shape functions in the `model.shape` field of the model object. The constructor of the Argyris shape function is `sharg_2_5`. The following properties are allowed:

TABLE 7-2: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHARG SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","sharg_2_5");
model.shape("shu").feature("f1").set("basename","u");
```

The Argyris element defines the following degrees of freedom (where *u* is the base name and *x* and *y* are the space coordinate names):

- *u* at corners
- *ux* and *uy* at corners, meaning derivatives of *u*
- *uxx*, *uxy*, and *uyy* at corners, meaning second derivatives
- *u*`n` at side midpoints, meaning a normal derivative. The direction of the normal is to the right if moving along an edge from a corner with lower mesh vertex number to a corner with higher number

The Argyris element defines the following field variables (where sdim = space dimension = 2 and edim = mesh element dimension):

- *u*
- *ux*, meaning the derivative of *u* with respect to *x*
- *uxy*, meaning a second derivative, defined for edim = sdim and edim = 0
- *uxTy*, the tangential derivative variable, meaning the *y*-component of the tangential projection of the gradient of *ux*, defined for 0 < edim < sdim

When calculating the derivatives, the global space coordinates are always expressed with shape order 1 in the Argyris element.

THE HERMITE ELEMENT

On each mesh element, the functions in the Hermite finite element space are the same as for the Lagrange element, namely, all polynomials of degree (at most) *k* in the local coordinates. The difference lies in which DOFs are used. For the Hermite element, a DOF `u` exists at each Lagrange point of order *k*, except at those points adjacent to a corner of the mesh element. These DOFs are the values of the function. In addition, other DOFs exist for the first derivatives of the function (with respect to the global coordinates) at the corners (`ux` and `uy` in 2D). Together, these DOFs determine the polynomials completely. The functions in the Hermite finite element space have continuous derivatives between mesh elements at the mesh vertices. However, at other common points for two mesh elements, these derivatives are not continuous. Thus, you can think of the Hermite element as lying between the Lagrange and Argyris elements.

The Hermite element is available with all types of mesh elements. The order $k \geq 3$ can be arbitrary, but the available numerical integration formulas usually limits its usefulness to $k \leq 5$ ($k \leq 4$ for tetrahedral meshes).

When setting Dirichlet boundary conditions on a variable that has Hermite shape functions, a locking effect can occur if the boundary is curved and the constraint order `cporder` is the same as the order of the Hermite element. This means that the derivative becomes over constrained at mesh vertices at the boundary, due to the implementation of the boundary conditions. To prevent this locking, you can specify `cporder` to be the element order minus 1.

Syntax for the Hermite Element (shherm)

Specify Hermite shape functions in the `model.shape` field of the model object. The constructor of the Hermite shape function is `shherm`. The following properties are allowed:

TABLE 7-3: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHHERM SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name
`order`	Integer >= 3		Basis function order
`border`	Integer		Alias for `order`
`sorder`	Positive integer	Determined by frame	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shherm");
model.shape("shu").feature("f1").set("order",3);
model.shape("shu").feature("f1").set("basename","u");
```

The Hermite element defines the following degrees of freedom:

- The value of the variable `basename` at each Lagrange node point that is not adjacent to a corner of the mesh element.
- The values of the first derivatives of `basename` with respect to the global space coordinates at each corner of the mesh element. The names of these derivatives are formed by appending the space coordinate names to `basename`.

The Hermite element defines the following field variables. Denote `basename` with u, and let x and y denote (not necessarily distinct) space coordinates. The variables are (sdim = space dimension and edim = mesh element dimension):

- u
- ux, meaning the derivative of u with respect to x, defined when edim = sdim or edim=0
- uxy, meaning a second derivative, defined when edim = sdim
- uTx, the tangential derivative variable, meaning the x-component of the tangential projection of the gradient, defined when 0 < edim < sdim
- $uTxy$, meaning xy-component of the tangential projection of the second derivative, defined when edim < sdim

When calculating the derivatives, the global space coordinates are expressed as polynomials of degree (at most) `sorder` in the local coordinates.

BUBBLE ELEMENTS

Bubble elements have shape functions that are zero on the boundaries of the mesh element and have a maximum in the middle of the mesh element. The shape function (there is only one for each mesh element) is defined by a lowest-order polynomial that is zero on the boundary of the element.

The bubble element are available with all types of mesh elements.

Syntax for Bubble Elements (shbub)

Specify Bubble shape functions in the `model.shape` field of the model object. The constructor of the Bubble shape function is `shbub`. The following properties are allowed:

TABLE 7-4: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHBUB SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	variable name		Base variable name
`mdim`	nonnegative integer	`sdim`	Dimension of the mesh elements on which the bubble exist
`sorder`	positive integer	determined by frame	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shbub");
model.shape("shu").feature("f1").set("mdim",2);
model.shape("shu").feature("f1").set("basename","u");
```

The bubble element has a single degree of freedom, `basename`, at the midpoint of the mesh element.

The bubble element defines the following field variables. Denote `basename` with u, and let x and y denote (not necessarily distinct) space coordinates. The variables are (sdim = space dimension and edim = mesh element dimension):

- u, defined when $\text{edim} \le \text{mdim}$, $u = 0$ if $\text{edim} < \text{mdim}$.
- ux, meaning the derivative of u with respect to x, defined when $\text{edim} = \text{mdim} = \text{sdim}$.

- uTx, the tangential derivative variable, meaning the x-component of the tangential projection of the gradient, defined when $\mathrm{mdim} < \mathrm{sdim}$ and $\mathrm{edim} \le \mathrm{mdim}$. $u\mathrm{T}x = 0$ if $\mathrm{edim} < \mathrm{mdim}$.
- uTxy, meaning the xy-component of the tangential projection of the second derivative, defined when $\mathrm{mdim} < \mathrm{sdim}$ and $\mathrm{edim} \le \mathrm{mdim}$. $u\mathrm{T}xy = 0$ if $\mathrm{edim} < \mathrm{mdim}$.

THE CURL ELEMENT

In electromagnetics, *curl elements* (also called *vector elements* or *Nédélec's edge elements*) are widely used. Each mesh element has DOFs corresponding only to tangential components of the field. For example, in a tetrahedral mesh in 3D each of the three edges in a triangle face element has degrees of freedom that are tangential components of the vector field in the direction of the corresponding edges, and in the interior there are degrees of freedom that correspond to vectors tangential to the triangle itself (if the element order is high enough). Finally, in the interior of the mesh tetrahedron there are degrees of freedom in all coordinate directions (if the element order is high enough). This implies that tangential components of the vector field are continuous across element boundaries, but the normal component is not necessarily continuous. This also implies that the curl of the vector field is an integrable function, so these elements are suitable for equations using the curl of the vector field.

The curl elements are available for all types of mesh elements. The polynomial order of the curl element can be at most three in 3D, and at most four in 2D and 1D.

Syntax for the Curl Element (shcurl)

Specify curl shape functions in the `model.shape` field of the model object. The constructor of the curl shape function is `shcurl`. The following properties are allowed:

TABLE 7-5: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHCURL SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`fieldname`	String		Field name
`compnames`	String array	Derived from fieldname	Names of vector field components
`dofbasename`	String	See below	Base name of degrees of freedom
`dcompnames`	String	See below	Names of the anti-symmetrized components of the gradient of the vector field
`order`	Integer		Basis function order

TABLE 7-5: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHCURL SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`border`	Positive integer	`order`	Alias for `order`
`sorder`	Positive integer	Given by `frame`	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shcurl");
model.shape("shu").feature("f1").set("order",2);
model.shape("shu").feature("f1").set("compnames",
            new String[]{"Ex","Ey"});
model.shape("shu").feature("f1").set("dofbasename","tE");
```

The default for `compnames` is `fieldname` concatenated with the space coordinate names. The default for `dofbasename` is `t`*`allcomponents`*, where *`allcomponents`* is the concatenation of the names in `compnames`.

The property `dcompnames` lists the names of the component of the antisymmetric matrix

$$dA_{ij} = \frac{\partial A_j}{\partial x_i} - \frac{\partial A_i}{\partial x_j},$$

where A_i are the vector field components and x_i are the space coordinates. The components are listed in row order. If a name is the empty string, the field variable corresponding to that component is not defined. If you have provided `compnames`, the default for the entries in `dcompnames` is `compnames(j) sdimnames(i) compnames(i) sdimnames(j)` for off-diagonal elements. If only `fieldname` has been given, the default for the entries are `dfieldname sdimnames(i)sdimnames(j)`. Diagonal elements are not defined per defaults. For example, `shcurl('order',3,'fieldname','A','dcompnames', {'','','curlAy','curlAz','','','','curlAx',''})`.

The curl element defines the following degrees of freedom: `dofbasename` d c, where $d = 1$ for DOFs in the interior of an edge, $d = 2$ for DOFs in the interior of a surface, etc., and c is a number between 0 and $d - 1$.

The curl element defines the following field variables (where comp is a component name from compnames, and dcomp is a component from dcompnames, sdim = space dimension and edim = mesh element dimension):

- comp, meaning a component of the vector, defined when edim = sdim.
- tcomp, meaning one component of the tangential projection of the vector onto the mesh element, defined when edim < sdim.
- comp*x*, meaning the derivative of a component of the vector with respect to global space coordinate *x*, defined when edim = sdim.
- tcompT*x*, the tangential derivative variable, meaning the *x* component of the projection of the gradient of tcomp onto the mesh element, defined when edim < sdim. Here, *x* is the name of a space coordinate.
- dcomp, meaning a component of the anti-symmetrized gradient, defined when edim = sdim.
- tdcomp, meaning one component of the tangential projection of the anti-symmetrized gradient onto the mesh element, defined when edim < sdim.

For performance reasons, prefer using dcomp in expressions involving the curl rather than writing it as the difference of two gradient components.

For the computation of components, the global space coordinates are expressed as polynomials of degree (at most) sorder in the local coordinates.

DISCONTINUOUS ELEMENTS

The functions in the discontinuous elements space are the same as for the Lagrange element, with the difference that the basis functions are discontinuous between the mesh elements.

Syntax for the Discontinuous Elements (shdisc and shhwdisc)

Specify discontinuous shape functions in the model.shape field of the model object. The constructor of the discontinuous shape functions is either shdisc or shhwdisc. The difference between these two shape function types is that the latter has optimal placement of degrees of freedom on triangle and tetrahedral meshes with respect to certain interpolation error estimates, whereas the former are available on all types of mesh elements with arbitrary polynomial order k. However, the available numerical integration formulas usually limits the usefulness to $k \leq 5$ ($k \leq 4$ for tetrahedral meshes). The following properties are allowed:

TABLE 7-6: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHDISC SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name
`order`	Integer		Basis function order
`mdim`	Nonnegative integer	`sdim`	Dimension of the mesh elements where the discontinuous element exists
`border`	Nonnegative integer	`order`	Alias for `order`
`sorder`	Positive integer	given by `frame`	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("sh1","frame1");
model.shape("sh1").feature().create("f1","shdisc");
model.shape("sh1").feature("f1").set("order",2);
model.shape("sh1").feature("f1").set("basename","u");
```

The discontinuous element defines the following field variables. Denote `basename` with u, and let x denote the space coordinates. The variables are (edim is the mesh element dimension):

- u, defined when edim = mdim.
- ux, meaning the derivative of u with respect to x, defined when edim = mdim = sdim.
- uTx, the tangential derivative variable, meaning the derivative of u with respect to x, defined when edim = mdim < sdim.

DENSITY ELEMENTS

The functions in the density elements space are the same as for the discontinuous element if the mesh element is not curved. If the element is curved, the functions define a density of the given order in *local* coordinates, and the value in global coordinates depends on the transformation between local and global coordinates.

The discontinuous elements are available with all types of mesh elements. The order k can be arbitrary, but the available numerical integration formulas usually limits its usefulness to $k \leq 5$ ($k \leq 4$ for tetrahedral meshes).

Syntax for the Density Element (shdens)

Specify density shape functions in the `model.shape` field of the model object. The constructor of the density shape function is `shdens`. The following properties are allowed:

TABLE 7-7: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHDENS SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name
`order`	Integer		Basis function order
`border`	Nonnegative integer	`order`	Alias for `order`
`sorder`	Positive integer	Given by `frame`	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shdens");
model.shape("shu").feature("f1").set("order",2);
model.shape("shu").feature("f1").set("basename","u");
```

The density element defines the following field variables. Denote `basename` with u, and let x denote the space coordinates. The variables are (edim is the mesh element dimension):

- u, defined when edim = sdim.
- ux, meaning the derivative of u with respect to x, defined when edim = sdim.

GAUSS POINT DATA ELEMENTS

The Gauss point data elements are not shape functions in the usual sense. Instead, they define discrete degrees of freedom associated with the Gauss points used in an integration rule of a given order. To make these degrees of freedom available for evaluation and use in equations, a field variable with the same name as the degree of freedom is also defined. This field can be evaluated at any point inside the element but simply evaluates to the degree of freedom value at the closest Gauss point.

Use the Gauss point data elements to, for example, store historical values at the integration points in models with hysteresis, or to solve an implicit equation for a material property at each point. Make sure to use elements of the same order as the integration rule used for assembly of the main PDE, typically twice the discretization order of the main dependent variables. This ensures that the Gauss point data really behaves as a truly local quantity without any spatial correlation.

Syntax for the Gauss Point Data Elements (shgp)

Specify Gauss point data shape functions in the `model.shape` field of the model object. The constructor of the density shape function is `shgp`. The following properties are allowed:

TABLE 7-8: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHDENS SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`basename`	Variable name		Base variable name
`order`	Integer		Basis function order
`mdim`	Integer		Element dimension
`valuetype`	real|complex	complex	Value type in case of using split representation of complex variables[a]

[a] The value type is ignored when split representation of complex variables is not used

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks. The following code creates a Gauss point data shape function declaring the degree of freedom u at integration points of order 4 in three-dimensional mesh elements.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shgp");
model.shape("shu").feature("f1").set("order",4);
model.shape("shu").feature("f1").set("basename","u");
model.shape("shu").feature("f1").set("mdim","3");
```

The Gauss point data element defines the following field variables. Denote `basename` with u and let edim be the evaluation dimension:

- u, defined when edim <= mdim.

DIVERGENCE ELEMENTS

For modeling the **B** (magnetic flux density) and **D** (electric displacement) fields in electromagnetics, the divergence elements are useful. The DOFs on the boundary of a mesh element correspond to normal components of the field. In addition, there are DOFs corresponding to all vector field components in the interior of the mesh element of dimension sdim (if the order is high enough). This implies that the normal component of the vector field is continuous across element boundaries, but the tangential components are not necessarily continuous. This also implies that the divergence of the vector field is an integrable function, so these elements are suitable for equations using the divergence of the vector field.

The divergence element are available with all types of mesh elements. The polynomial order of the divergence element can be at most three in 3D, and at most four in 2D and 1D.

Syntax for Divergence Elements (shdiv)

Specify divergence shape functions in the `model.shape` field of the model object. The constructor of the divergence shape function is `shdiv`. The following properties are allowed:

TABLE 7-9: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHDIV SHAPE FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
`fieldname`	Variable name		Name of vector field
`compnames`	String array	Derived from fieldname	Names of vector field components
`dofbasename`	String	see below	Base name of degrees of freedom
`divname`	String	see below	Name of divergence field
`order`	Integer	1	Basis function order
`border`	Positive integer	`order`	Alias for `order`
`sorder`	Positive integer	Given by `frame`	Geometry shape order

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shdiv");
model.shape("shu").feature("f1").set("order",2);
model.shape("shu").feature("f1").set("compnames",
            new String[]{"Bx","By"});
model.shape("shu").feature("f1").set("dofbasename","nB");
```

The default for `compnames` is `fieldname` concatenated with the space coordinate names. The default for `dofbasename` is `n`*`allcomponents`*, where *`allcomponents`* is the concatenation of the names in `compnames`.

The vector element defines the following degrees of freedom: `dofbasename` on element boundaries, and `dofbasename` sdim c, $c = 0, \ldots,$ sdim $- 1$ for DOFs in the interior.

The divergence element defines the following field variables (where comp is a component name from compnames, divname is the divname, sdim = space dimension and edim = mesh element dimension):

- comp, meaning a component of the vector, defined when edim = sdim.
- ncomp, meaning one component of the projection of the vector onto the normal of mesh element, defined when edim = sdim − 1.
- comp*x*, meaning the derivative of a component of the vector with respect to global space coordinate *x*, defined when edim = sdim.
- ncompT*x*, the tangential derivative variable, meaning the *x* component of the projection of the gradient of ncomp onto the mesh element, defined when edim < sdim. Here, *x* is the name of a space coordinate. ncompT*x* = 0.
- divname, means the divergence of the vector field.

For performance reasons, prefer using divname in expressions involving the divergence rather than writing it as the sum of sdim gradient components.

For the computation of components, the global space coordinates are expressed as polynomials of degree (at most) sorder in the local coordinates.

SCALAR PLANE WAVE BASIS FUNCTION

The scalar plane wave basis function, shuwhelm, is used to implement scalar plane wave basis functions for solving scalar wave equations of Helmholtz type using an *ultraweak variational formulation* (UWVF). These basis functions are discontinuous in between mesh elements.

Syntax for the Scalar Plane Wave Basis Function (shuwhelm)

Specify scalar plane wave basis functions in the model.shape field of the model object. The constructor of the density shape function is shuwhelm. The following properties are allowed:

TABLE 7-10: VALID PROPERTY NAME/VALUE PAIRS FOR THE SHUWHELM WAVE BASIS FUNCTION

PROPERTY	VALUE	DEFAULT	DESCRIPTION
basename	Variable name		Base variable name
ndir	Integer		Number of wave directions
kexpr	String		Variable for the wave number
xexpr	String array	{'x','y','z'}	Expressions for the x, y, and z coordinate transformations

The property names cannot be abbreviated and must be written in lowercase letters enclosed in quotation marks.

```
model.shape().create("shu","f");
model.shape("shu").feature().create("f1","shuwhelm");
model.shape("shu").feature("f1").set("ndir",2);
model.shape("shu").feature("f1").set("basename","u");
```

The scalar plane wave basis function defines the following field variables. Denote `basename` with u, and let x denote the space coordinates. The variables are (sdim = space dimension):

- u
- ux, meaning the derivative of u with respect to x, defined on edim = sdim

Discretization of the Equations

This section describes how COMSOL Multiphysics forms the discretization of the PDE. Consider a 2D problem for simplicity. The starting point is the weak formulation of the problem. First comes the discretization of the constraints

$$0 = R^{(2)} \text{on } \Omega$$
$$0 = R^{(1)} \text{on B}$$
$$0 = R^{(0)} \text{on P}$$

starting with the constraints on the boundaries, B. For each mesh element in B (that is, each mesh edge in B), consider the Lagrange points of some order k (see The Lagrange Element). Denote them by $x_{mj}^{(1)}$, where m is the index of the mesh element. Then the discretization of the constraint is

$$0 = R^{(1)}(x_{mj}^{(1)}),$$

that is, the constraints must hold pointwise at the Lagrange points. The Lagrange point order k can be chosen differently for various components of the constraint vector $R^{(1)}$, and it can also vary in space. COMSOL Multiphysics' data structures denote the k as `cporder`. The constraints on domains Ω and points P are discretized in the same way. (Nothing needs to be done with the points P.) You can collect all these pointwise constraints in one equation $0 = M$, where M is the vector consisting of all the right-hand sides.

COMSOL Multiphysics approximates the dependent variables with functions in the chosen finite element space(s). This means that the dependent variables are expressed in terms of the degrees of freedom as

$$u_l = \sum_i U_i \varphi_i^{(l)}$$

where $\varphi_i^{(l)}$ are the basis functions for variable u_l. Let U be the vector with the degrees of freedoms U_i as its components. This vector is called the *solution vector* because it is what you want to compute. M depends only on U, so the constraints can be written $0 = M(U)$.

In case of complex variables the default representation of the complex variables is to use complex degrees of freedom U_i. An alternative possibility is to use split representation where the complex variables is represented using pairs of real degrees of freedom as

$$u_l = \sum_k U_k \varphi_k^{(l)} + i V_k \varphi_k^{(l)}$$

In this case the solution vector has U_k and V_k as its components.

Now consider the weak equation:

$$0 = \int_\Omega W^{(2)} dA + \int_B W^{(1)} ds + \sum_P W^{(0)}$$
$$-\int_\Omega v \cdot h^{(2)^T} \mu^{(2)} dA - \int_B v \cdot h^{(1)^T} \mu^{(1)} ds - \sum_P v \cdot h^{(0)^T} \mu^{(0)}$$

where $\mu^{(i)}$ are the Lagrange multipliers. To discretize it, express the dependent variables in terms of the DOFs as described earlier. Similarly, approximate the test functions with the same finite elements (this is the Galerkin method):

$$v_l = \sum_i V_i \varphi_i^{(l)}$$

Because the test functions occur linearly in the integrands of the weak equation, it is enough to require that the weak equation holds when you choose the test functions as basis functions:

$$v_l = \varphi_i^{(l)}$$

When substituted into the weak equation, this gives one equation for each i. Now the Lagrange multipliers must be discretized. Let

$$\Lambda_{mj}^{(d)} = \mu^{(d)}(x_{mj}^{(d)})w_{mj}^{(d)}$$

where $x_{mj}^{(d)}$ are the Lagrange points defined earlier, and $w_{mj}^{(d)}$ are certain weights (see the following discussion). The term

$$\int_B \varphi_i \cdot h^{(1)^T} \mu^{(1)} ds$$

is approximated as a sum over all mesh elements in B. The contribution from mesh element number m to this sum is approximated with the Riemann sum

$$\sum_j \varphi_i(x_{mj}^{(1)}) \cdot h^{(1)^T}(x_{mj}^{(1)})\mu^{(1)}(x_{mj}^{(1)})w_{mj}^{(1)} = \sum_j \varphi_i(x_{mj}^{(1)}) \cdot h^{(1)^T}(x_{mj}^{(1)})\Lambda_{mj}^{(1)}$$

where $w_{mj}^{(1)}$ is the length (or integral of ds) over the appropriate part of the mesh element. The integral over Ω and the sum over P is approximated similarly.

All this means that you can write the discretization of the weak equation as

$$0 = L - N_F\Lambda$$

where L is a vector whose ith component is

$$\int_\Omega W^{(2)} dA + \int_B W^{(1)} ds + \sum_P W^{(0)}$$

evaluated for $v_l = \varphi_i^{(l)}$. Λ is the vector containing all the discretized Lagrange multipliers $\Lambda_{mj}^{(d)}$. N_F is a matrix whose ith row is a concatenation of the vectors

$$\varphi_i(x_{mj}^{(d)})h^{(d)}(x_{mj}^{(d)})^T$$

For problems using *ideal constraints*, N_F is equal to the *constraint Jacobian matrix* N, which is defined as

$$N = -\frac{\partial M}{\partial U}$$

To sum up, the discretization of the stationary problem is

$$\begin{aligned} 0 &= L(U)-N_F(U)\Lambda \\ 0 &= M(U) \end{aligned}$$

The objective is to solve this system for the solution vector U and the Lagrange multiplier vector Λ. L is called the *residual vector*, M is the *constraint residual*, and N_F is the *constraint force Jacobian matrix.* M is redundant in the sense that some pointwise constraints occur several times. Similarly, Λ is redundant. Solvers remove this redundancy.

NUMERICAL QUADRATURE

The integrals occurring in the components of the residual vector L (as well as K, as noted later in this discussion) are computed approximately using a *quadrature formula*. Such a formula computes the integral over a mesh element by taking a weighted sum of the integrand evaluated in a finite number of points in the mesh element. The *order* of a quadrature formula on a 1D, triangular, or tetrahedral element is the maximum number k such that it exactly integrates all polynomials of degree k. For a quadrilateral element, a formula of order k integrates exactly all products $p(\xi_1)q(\xi_2)$, where p and q are polynomials of degree k in the first and second local coordinates, respectively. A similar definition holds for hexahedral and prism elements. Thus the accuracy of the quadrature increases with the order. On the other hand, the number of evaluation points also increases with the order. As a rule of thumb, you can take the order to be twice the order of the shape function for the finite element being used. COMSOL Multiphysics' data structures refer to the order of the quadrature formula as `gporder` (`gp` stands for Gauss points). The maximum available order of the quadrature formula (the `gporder` value) is:

- 41 for 1D, quadrilateral, and hexahedral meshes
- 30 for triangular and prism meshes
- 8 for tetrahedral meshes

TIME-DEPENDENT PROBLEMS

The discretization of a time-dependent problem is similar to the stationary problem

$$\begin{aligned} 0 &= L(U, \dot{U}, \ddot{U}, t)-N_F(U, t)\Lambda \\ 0 &= M(U, t) \end{aligned}$$

where U and Λ now depend on time t.

LINEARIZED PROBLEMS

Consider a linearized stationary problem. The linearization "point" u_0 corresponds to a solution vector U_0. The discretization of the linearized problem is

$$\begin{gathered} K(U_0)(U-U_0)+N_F(U_0)\Lambda = L(U_0) \\ N(U_0)(U-U_0) = M(U_0) \end{gathered}$$

where K is called the *stiffness matrix*, and $L(U_0)$ is the *load vector*. For problems given in general or weak form, K is the Jacobian of $-L$:

$$K = -\frac{\partial L}{\partial U}$$

The entries in the stiffness matrix are computed in a similar way to the load vector, namely by integrating certain expressions numerically. This computation is called the *assembling* the stiffness matrix.

If the original problem is linear, then its discretization can be written

$$\begin{gathered} KU+N_F\Lambda = L(0) \\ NU = M(0) \end{gathered}$$

Similarly, for a time-dependent model the linearization involves the *damping matrix*

$$D = -\frac{\partial L}{\partial \dot{U}}$$

and the *mass matrix*

$$E = -\frac{\partial L}{\partial \ddot{U}}$$

When $E = 0$, the matrix D is often called the mass matrix instead of the damping matrix.

EIGENVALUE PROBLEMS

The discretization of the eigenvalue problem is

$$\begin{gathered} \lambda^2 E(U_0)U - \lambda D(U_0)U + K(U_0)U + N_F(U_0)\Lambda = 0 \\ N(U_0)U = 0 \end{gathered}$$

where U_0 is the solution vector corresponding to the linearization "point." If the underlying problem is linear, then D, K, and N do not depend on U_0, and you can write

$$KU + N_F\Lambda = \lambda DU - \lambda^2 EU$$
$$NU = 0$$

WEAK CONSTRAINTS

Weak constraints present an alternative way to discretize the Dirichlet conditions, as opposed to the pointwise constraints described earlier. The idea is to regard the Lagrange multipliers $\mu^{(d)}$ as field variables and thus approximate them with finite elements. This concept also introduces corresponding test functions $\nu^{(d)}$. Multiply the Dirichlet conditions with these test functions and integrate to end up with the following system in the case of a stationary problem in 2D:

$$0 = \int_{\Omega} W^{(2)} dA + \int_{B} W^{(1)} ds + \sum_{P} W^{(0)}$$
$$-\int_{\Omega} v \cdot h^{(2)T} \mu^{(2)} dA - \int_{B} v \cdot h^{(1)T} \mu^{(1)} ds - \sum_{P} v \cdot h^{(0)T} \mu^{(0)}$$
$$0 = \int_{\Omega} \nu^{(2)} \cdot R^{(2)} dA$$
$$0 = \int_{B} \nu^{(1)} \cdot R^{(1)} ds$$
$$0 = \sum_{P} \nu^{(0)} \cdot R^{(0)}$$

You could add these weak equations to form a single equation. This treatment of the Lagrange multipliers as ordinary variables has thus produced a weak equation without constraints. This can be useful if the Lagrange multipliers are of interest in their own right.

Take care when combining pointwise and weak constraints. For instance, if you have both types of constraints for some variable and the constraints are in adjacent domains, the resulting discretization does not work. You can obtain pointwise constraints from the weak constraints formulation by using the basis functions

$$\delta(x - x_{mj}^{(d)})$$

for the Lagrange multipliers and their test functions, that is, let

$$\mu^{(d)} = \sum_{m,j} \Lambda_{mj}^{(d)} \delta(x - x_{mj}^{(d)})$$

where δ is Dirac's delta function.

What Equations Does COMSOL Multiphysics Solve?

This section explains how COMSOL Multiphysics forms the system of equations and constraints that it solves. It also discusses the implications of Dirichlet conditions involving several solution components in a multiphysics model.

You specify material parameters and boundary conditions using features in a number of *physics interfaces*. Each physics interface forms one or several PDEs and boundary conditions from these settings. If you use one of the interfaces for equation-based modeling, you specify the equation coefficients and terms directly.

The software collects all the equations and boundary conditions formulated by the physics interfaces into one large system of PDEs and boundary conditions. COMSOL Multiphysics solves the system using a weak formulation. In this section The PDE Forms are described.

The PDE Forms

COEFFICIENT FORM

In the coefficient equation system form, the PDEs and boundary conditions are written in following form (for a time-dependent model):

$$\begin{cases} e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au=f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)+qu=g-h^T\mu & \text{on } \partial\Omega \\ hu=r & \text{on } \partial\Omega \end{cases}$$

Enter equations on coefficient form by selecting the **Coefficient Form PDE** interface in the **Model Wizard**.

GENERAL FORM

In the general equation system form, the PDEs and boundary conditions are written in following form (for a time-dependent model):

$$\begin{cases} e_a\ddot{u} + d_a\dot{u} + \nabla\cdot\Gamma = F & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = G - h^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

Enter equations on coefficient form by selecting the **General Form PDE** interface in the **Model Wizard**.

WAVE FORM

In the wave equation system form, the PDEs and boundary conditions are written in following form:

$$\begin{cases} d_a\dot{u} + \nabla\cdot\Gamma = f & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = G & \text{on } \partial\Omega \\ u = r & \text{on } \partial\Omega \end{cases}$$

Enter equations on coefficient form by selecting the **Wave Form PDE** interface in the **Model Wizard**.

Note

The wave form is somewhat special regarding the underlying discretization in space and time.

WEAK FORM

Note

The weak form is described in detail in Weak Form Theory Background. Most physics interfaces implement their equations in the weak form.

All equations specify are converted to the weak form before solving. To see the weak equations derived, click the **Show** button ( ) and select **Equation View** and click on the **Equation View** nodes that appear under the nodes in the Model Builder.

The full system of equations and constraints is approximated using the finite element method; see Discretization of the Equations.

Weak Form Theory Background

The mathematical weak form gives you direct access to the terms of the weak equation and provides maximum freedom in defining finite element problems. This section provides a theoretical background to the weak form in COMSOL Multiphysics.

In this section:

- Deriving the Weak Form
- Entering a PDE in Coefficient Form Using the Weak Form

Deriving the Weak Form

Before describing what the weak form is, this discussion first looks at the conversion of a simple problem in general form to the weak form.

Consider a stationary PDE problem for a single dependent variable, u, in two space dimensions:

$$\nabla \cdot \Gamma = F \qquad \text{in } \Omega$$

Now let v be an arbitrary function on Ω, and call it the *test function* (v should of course belong to a suitably chosen well-behaved class of functions, V). Multiplying the PDE with this function and integrating leads to

$$\int_{\Omega} v\nabla \cdot \Gamma dA = \int_{\Omega} vFdA$$

where dA is the area element. Now use Green's formula (Gauss' formula) to integrate by parts:

$$\int_{\partial\Omega} v\Gamma \cdot \mathbf{n}ds - \int_{\Omega} \nabla v \cdot \Gamma dA = \int_{\Omega} vFdA$$

where ds is the length element. Then use the Neumann boundary condition

$$-\mathbf{n} \cdot \Gamma = G + \frac{\partial R}{\partial u}\mu$$

to arrive at the following equation:

$$0 = \int_{\Omega} (\nabla v \cdot \Gamma + vF) dA + \int_{\partial\Omega} v\left(G + \frac{\partial R}{\partial u}\mu\right) ds$$

Together with the Dirichlet condition, this is a weak reformulation of the original PDE problem. The requirement is that the above weak equation must hold for *all* test functions v. You can reverse the steps of the derivation to show that if the functions u and μ satisfy the weak formulation, then they also satisfy the original formulation. However, this holds true only if the solutions and coefficients are sufficiently smooth. For instance, in the case of discontinuities in material properties, you can have a solution of the weak formulation while the strong formulation has no meaning. The names "weak" and "strong" stem from this distinction: the weak formulation is a weaker condition on the solution than the strong formulation. A benefit of the weak formulation is that it requires less regularity of Γ. This is important in the finite element method.

For the most part, the weak reformulation of a system of equations in general form is

$$\begin{cases} 0 = \int_{\Omega} (\nabla v_l \cdot \Gamma_l + v_l F_l) dA + \int_{\partial\Omega} v_l\left(G_l + \frac{\partial R_m}{\partial u_l}\mu_m\right) ds \\ 0 = R_m \quad \text{on } \partial\Omega \end{cases}$$

where you use the summation convention. If the boundary conditions are given on interior boundaries, $\partial\Omega$ can be enlarged to include these. You now have several test functions $v_1, v_2, \ldots, v_N$, one for each of the original PDEs. The weak equation is required to hold for all test functions (in the suitably chosen function space V). Of course, in three dimensions, for dA and ds you must substitute the volume and surface area elements, respectively. In one dimension, for dA substitute the length element and interpret the boundary integral as a sum over the boundary points.

VARIATIONAL PRINCIPLES

This section provides some background on the weak formulation. You can eliminate the Lagrange multipliers from the weak equation in the previous section by introducing the following boundary conditions on the test functions:

$$0 = \frac{\partial R_m}{\partial u_l} v_l \quad \text{on } \partial\Omega$$

This means that the functions $u_l + v_l$ satisfy the linearized version of the Dirichlet boundary conditions. Then the weak formulation becomes the following: find functions u_l such that

$$\begin{cases} 0 = \int\limits_{\Omega} (\nabla v_l \cdot \Gamma_l + v_l F_l) dA + \int\limits_{\partial\Omega} v_l G_l ds \\ 0 = R_m \qquad \text{on } \partial\Omega \end{cases}$$

holds for all test functions v_l that satisfy the above boundary condition. Such a formulation arises when you have a *variational principle*; for instance, to find the functions u_l that minimizes the energy of some physical system under the constraints $0 = R_m$. If the energy is given as an integral of some expression involving the u_l, then the stationarity condition on the solution is precisely the weak formulation above. Because variational principles are more fundamental than the corresponding PDE, the weak form is often more natural than the strong forms. The connection to variational principles also shows why the term involving the Lagrange multipliers takes the form it does (that is, how ideal constraints come about).

THE WEAK FORM PDE STRUCTURE

Now consider what a problem on the weak form can look like. It can be more general than the weak equations derived from the general form above because it also contains integrals over edges (in 3D) or points (in 2D and 3D). Similarly, you can also have constraints on domains, edges, and points. Further, expressions occurring in the integrands can be more or less arbitrary functions of the variables u_l, the test functions v_l, and their derivatives. One important restriction, though, is that the test functions enter linearly. A problem in weak form has the following structure (in 2D):

$$0 = \int\limits_{\Omega} (W^{(2)} - W^{(2t)}) dA + \int\limits_{B} (W^{(1)} - W^{(1t)}) ds + \sum_{P} (W^{(0)} - W^{(0t)}) +$$

$$+ \int\limits_{\Omega} v_l \frac{\partial R_m^{(2)}}{\partial u_l} \mu_m^{(2)} dA + \int\limits_{B} v_l \frac{\partial R_m^{(1)}}{\partial u_l} \mu_m^{(1)} ds + \sum_{P} v_l \frac{\partial R_m^{(0)}}{\partial u_l} \mu_m^{(0)}$$

$$0 = R^{(2)} \qquad \text{on } \Omega$$

$$0 = R^{(1)} \qquad \text{on B}$$

$$0 = R^{(0)} \qquad \text{on P}$$

The first equation is the weak equation, and the others are the constraints. Here Ω is the domain, B is the boundary (including $\partial\Omega$ and interior boundaries), and P represents the vertices (points) defined in the geometry. The integrands $W^{(n)}$, which correspond to the sum of all **weak** field entries for dimension n, are scalar expressions involving the dependent variables u_1, u_2,..., u_N as well as the test functions v_1, v_2,..., v_N and their derivatives. You must enter the test functions and their derivatives linearly. For instance, you can have

$$W^{(2)} = v_1\frac{\partial u_2}{\partial x} + \frac{\partial v_2}{\partial x}\sin u_1 + \frac{\partial^2 v_1}{\partial y^2}$$

The quantities $R^{(i)}$ occurring in the constraints are vector-valued functions of the dependent variables. Sometimes they also depend on the derivatives of the dependent variables. In such cases there are additional terms accounting for this dependence in the integrals with Lagrange multipliers. You now also can have Lagrange multipliers on domains ($\mu^{(2)}$) and on vertices ($\mu^{(0)}$).

In 3D, contributions to the weak equation and the constraints can also come from edges (curves).

CONVERSION TO WEAK FORM

COMSOL Multiphysics converts equations in coefficient form or general form to weak form conceptually in the finite element method. The following equations describe the obtained weak form problem as well as problems given directly in weak form. For simplicity this discussion considers a stationary problem in 2D.

$$\begin{aligned}0 = &\int_\Omega (W^{(2)} - W^{(2t)})dA + \int_B (W^{(1)} - W^{(1t)})ds + \sum_P (W^{(0)} - W^{(0t)})\\ &-\int_\Omega v \cdot h^{(2)^T}\mu^{(2)}dA - \int_B v \cdot h^{(1)^T}\mu^{(1)}ds - \sum_P v \cdot h^{(0)^T}\mu^{(0)}\end{aligned}$$

$$0 = R^{(2)} \quad \text{on } \Omega$$

$$0 = R^{(1)} \quad \text{on B}$$

$$0 = R^{(0)} \quad \text{on P}$$

Here v is a vector whose components are the test functions v_l, while $\mu^{(d)}$ are column vectors of Lagrange multipliers $\mu_m{}^{(d)}$. The elements of matrices $h^{(d)}$ are

$$h_{ml}{}^{(d)} = -\frac{\partial R_m{}^{(d)}}{\partial u_l}$$

for a problem given directly in weak form. For a problem originally given in coefficient form or general form, the vectors $R^{(2)}$ and $R^{(0)}$ as well as the Lagrange multipliers $\mu^{(2)}$ and $\mu^{(0)}$ are empty. In addition, $W^{(0)} = 0$. For the general form you have

$$h_{lm}{}^{(1)} = -\frac{\partial R_m}{\partial u_l}$$

For the coefficient form, you have $h^{(1)} = h$.

For a problem given directly in weak form, the constraints can also depend on variables other than the u_l (for instance the space derivatives of u_l). In such cases, the quantities $v \cdot h^{(d)^T}$ are augmented with additional terms that account for this dependence.

Conversion from general form to weak form is performed according to

$$W^{(n)} = \Gamma_{lj}\frac{\partial v_l}{\partial x_j} + F_l v_l + \sum_l W_l^{0(n)}$$

$$W^{(nt)} = d_{alk}\frac{\partial u_k}{\partial t}v_l + \sum_l W_l^{0(nt)}$$

$$W^{(n-1)} = G_l v_l + \sum_l W_l^{0(n-1)}$$

$$R_m^{(n)} = R_m$$

where there is an implicit summation over repeated indices in each product. The superscript (n) is the space dimension, while the superscript 0 marks explicit weak contributions added to the general form equation. Explicit time derivative terms for historical reasons are collected in the $W^{(nt)}$. There is nothing that prevents you from entering time derivatives in $W^{(n)}$, though.

Entering a PDE in Coefficient Form Using the Weak Form

Derive the weak equation for a coefficient form problem:

$$0 = \int_{\Omega}(\nabla v\cdot(-c\nabla u-\alpha u+\gamma)+v(f-\beta\cdot\nabla u-au))dA+\int_{\partial\Omega}v(-qu+g-h^T\mu)ds$$

Suppose *c* is a scalar. Using the weak form for a 2D problem with a dependent variable `u`, follow these steps:

1. In a field for a weak expression in the **Settings** windows for a **Weak Form PDE** node or a **Weak Contribution** node on a domain, type

```
test(ux)*(-c*ux-alx*u+gax)+test(uy)*(-c*uy-aly*u+gay)+...
    test(u)*(f-bex*ux-bey*uy-a*u)
```

2. In a field for a weak expression in the **Settings** windows for a **Weak Form PDE** node or a **Weak Contribution** node on a boundary, type

```
test(u)*(-q*u+g)
```

The variables *alx*, *aly*, *bex*, *bey*, *gax*, and *gay* represent the components of α, β, and γ, respectively. You must replace the coefficients in this weak form expression (for example, *c*, *alx*, and *q*) with appropriate expressions.

COMSOL Multiphysics automatically adds the Lagrange multiplier μ.

8

Stabilization Techniques

Numerical solutions of transport equations can sometimes exhibit oscillations even when the exact solutions are smooth. These spurious oscillations are caused by numerical instabilities.

The first section describes when spurious oscillations can arise. To restore numerical stability COMSOL Multiphysics provides a set of stabilization techniques, which are described in the second section.

This chapter does not contain description of the user interface, which is instead described in conjunction with each physics interface that supports any stabilization technique.

Numerical Stabilization

In this section:

- About Numerical Stabilization in COMSOL Multiphysics
- An Example of Stabilization

About Numerical Stabilization in COMSOL Multiphysics

This section discusses the numerical stability of the generic scalar convection-diffusion transport equation

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + F \tag{8-1}$$

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 8-1 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):

$$\mathrm{Pe} = \frac{\|\beta\| h}{2c} > 1 \tag{8-2}$$

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 non-constant source term or a non-constant 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 adds artificial diffusion. COMSOL Multiphysics provides several of these methods and described in An Example of Stabilization.

An Example of Stabilization

This example uses the Heat Transfer interface.

To illustrate the concepts of this section, consider the problem

$$\cos\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial x} + \sin\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial y} = 10^{-4}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + 1 \qquad (8\text{-}3)$$

solved on the unit square. Equation 8-3 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 8-1 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 8-2 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.

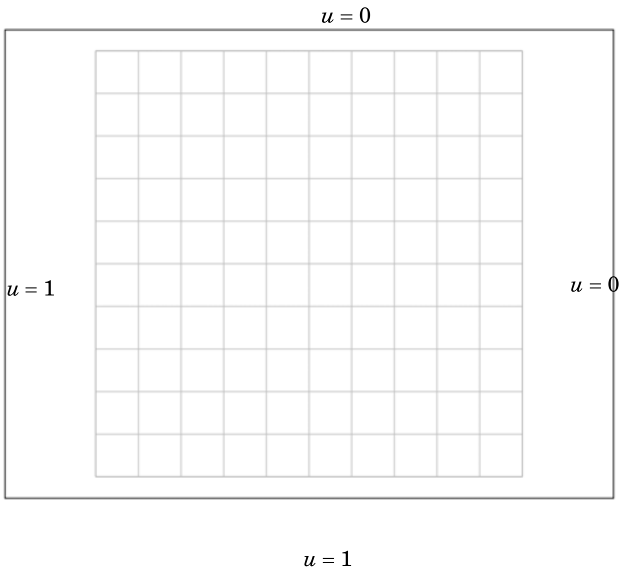


Figure 8-1: The computation domain, mesh, and boundary condition for Equation 8-3.

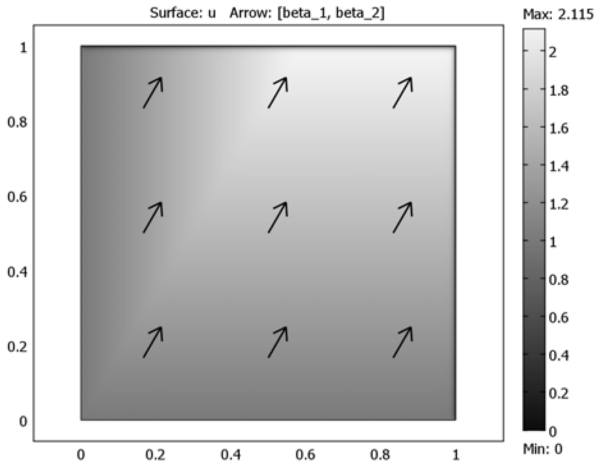


Figure 8-2: Reference solution of Equation 8-3*. Solved using 100 times 100 biquadratic elements with streamline diffusion and crosswind diffusion.*

The cell Péclet number for this example is

$$\mathrm{Pe} = \frac{1 \cdot 0.1}{2 \cdot 10^{-4}} = 500 >> 1$$

Figure 8-3 displays the solution obtained using the mesh in Figure 8-1 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 8-2. The right plot in Figure 8-3 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.

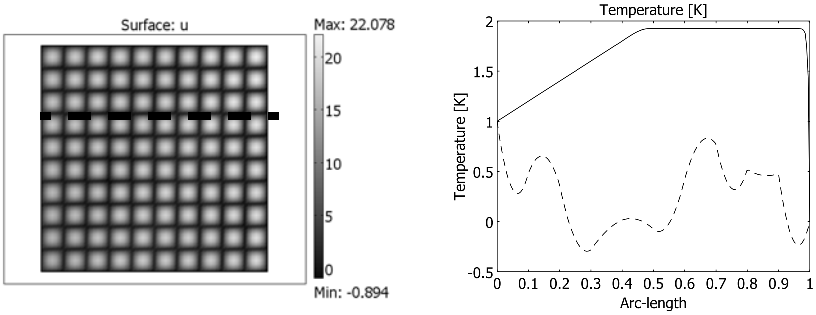


Figure 8-3: Equation 8-3 solved using unstabilized Galerkin formulation. The right plot compares the unstabilized solution (dashed line) along the dashed line in the left plot ($y = 0.8$) with the reference solution (solid line).

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 also common names) that stabilize the solution. To display these sections, click the **Show** button ( ) and select **Stabilization**.

- Consistent Stabilization
- Inconsistent Stabilization
- Isotropic Diffusion
- Streamline Diffusion
- Crosswind Diffusion
- References for Stabilization Techniques

See Also

In the *COMSOL Multiphysics User's Guide:*

- Show Stabilization
- Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces
- Numerical Stability—Stabilization Techniques for the Fluid Flow Interfaces

Consistent Stabilization

A consistent stabilization method adds numerical diffusion in such a way that if u is an exact solution to Equation 8-1, 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 8-1, 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 8-1, but rather, the modified $O(h)$ perturbed problem

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot ((c + c_{\mathrm{art}}) \nabla u) + F \tag{8-4}$$

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 it should be chosen 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 8-4 shows the effect of isotropic diffusion on Equation 8-3 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.

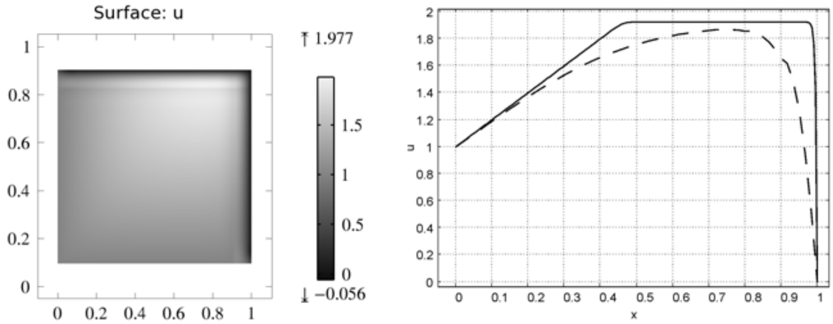


Figure 8-4: Equation 8-3 *solved using isotropic diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).*

Streamline Diffusion

The streamline diffusion method in COMSOL is a consistent stabilization method. When applied to Equation 8-1, 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 8-5 displays the effect of SUPG on the solution of Equation 8-3. 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.

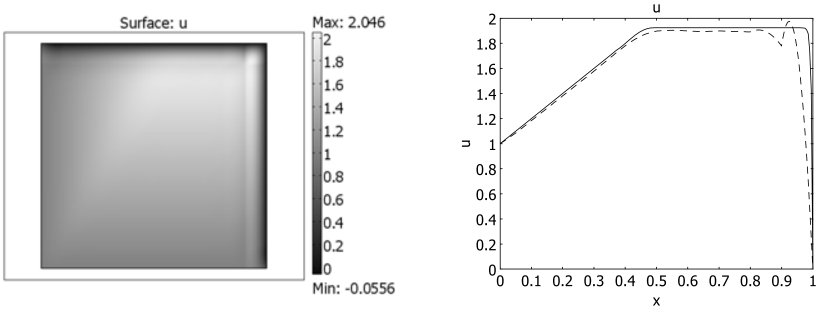


Figure 8-5: Equation 8-3 *solved using streamline diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).*

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 8-1:

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + su + F \tag{8-5}$$

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 8-5 is characterized by the Péclet number (see Equation 8-2) 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:

$$2\mathrm{DaPe} = \frac{|s|h^2}{c} \tag{8-6}$$

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 provided that the exact solution of Equation 8-1 (or Equation 8-5) does not contain any discontinuities. At sharp gradients, however, undershoots and overshoots can occur in the numerical solutions as can be seen in Figure 8-5. Crosswind diffusion addresses these spurious oscillations by adding diffusion orthogonal to the streamline direction, that is, in the crosswind direction.

The crosswind diffusion methods is consistent, but it is nonlinear. This means that the discrete equation system becomes nonlinear even if the original equation (Equation 8-1 or Equation 8-5) is linear, which can increase the computational cost.

Note

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 not be greater than one or smaller than zero.

Ref. 5 presents a comprehensive review of most of the existing crosswind diffusion methods. Which of these methods that is used depend on the physics interface.

Figure 8-6 shows the example problem (Equation 8-3) solved using streamline diffusion and crosswind diffusion. Oscillations at the boundary layers are almost completely removed (compare with Figure 8-5), 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 8-1). To obtain a sharper solution and remove the last oscillations, the mesh needs to be refined locally at the boundary layers.

[1] The streamline diffusion stabilization in COMSOL is GLS but without any viscous terms in the test operator in the stabilization term.

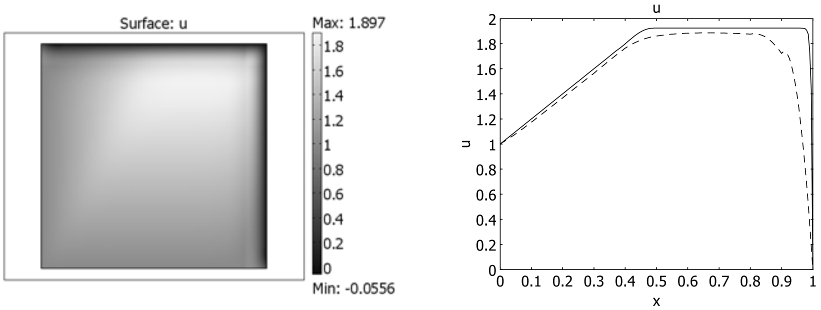


Figure 8-6: Equation 8-3 *solved using streamline diffusion and crosswind diffusion. The right plot compares the stabilized solution (dashed line) along y = 0.8 with the reference solution (solid line).*

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. V. John and P. Knobloch, “On Spurious Oscillations at Layers Diminishing (SOLD) Methods for Convection-Diffusion Equations: Part I - A Review,” *Comput. Methods Appl. Mech. Engrg.*, vol. 196, pp. 2197–2215, 2007.

9

Advanced Solver Topics

This chapter examines some of the solvers and solver algorithms in COMSOL Multiphysics in some detail.

In this chapter:

Solver Algorithms

In this section:

- The Nonlinear Solver Algorithm
- The Time-Dependent Solver Algorithm
- The Eigenvalue Solver Algorithm
- The Parametric Solver Algorithm
- The Stationary Segregated Solver Algorithm
- The Adaptive Solver Algorithm
- The Sensitivity Solver Algorithm
- References for Solver Algorithms

See Also

- Solver Features
- Solvers and Study Types in the *COMSOL Multiphysics User's Guide*
- Solver API in the *COMSOL Java API Reference Guide*
- The *Optimization Module User's Guide* for information about the solver's for this add-on product.

The Nonlinear Solver Algorithm

The nonlinear solver uses an affine invariant form of the damped Newton method as described in Ref. 1. 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 code 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. Thus you should spend considerable time providing the best value for U_0, giving at least an order of magnitude guess for different solution components.

CONVERGENCE CRITERION

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

$$\mathrm{err} = \left(\frac{1}{N}\sum_{i=1}^{N}(|E_i|/W_i)^2\right)^{1/2}$$

Here N is the number of degrees of freedom and $W_i = \max(|U_i|, S_i)$, where S_i is a scale factor that the solver determines on the basis of the scaling method, which you can select from the **Method** list in the **Scaling** section of the **Variables** node's **Settings** window, according to the following rules:

- For **Automatic**, S_i is the average of $|U_j|$ for all DOFs j having the same name as DOF i times a factor equal to 10^{-5} for highly nonlinear problems or 0.1 otherwise.
- For **Manual**, S_i is the value for DOF i given in the **Scale** field.
- For **Initial value based**, S_i is the factor s (see below) times the average of $|U_{0j}|$ for all DOFs j having the same name as DOF i, where U_0 is the solution vector corresponding to the initial value.
- For **None**, $W_i = 1$. In this case, err is an estimate for the absolute error.

Note

The 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.

The Time-Dependent Solver Algorithm

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 Multiphysics, the two solvers IDA and Generalized-α are available to solve the above ODE or DAE system.

IDA was created at the Lawrence Livermore National Laboratory (see Ref. 2) and is, in turn, a modernized implementation of the DAE solver DASPK (see Ref. 3), which uses variable-order variable-step-size backward differentiation formulas (BDF).

Generalized-α is an implicit, second-order accurate method with a parameter called α or ρ_∞, where $0 \le \rho_\infty \le 1$, to control how much high frequencies are damped. With $\rho_\infty = 1$, the method has no numerical damping. For linear problems, this corresponds to the midpoint rule. The choice $\rho_\infty = 0$ 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 (see Ref. 4) and later extended to first-order systems (see Ref. 5).

Because the time-stepping schemes used are implicit, a possibly nonlinear system of equations must be solved each time step. Included in IDA is a Newton solver to solve this nonlinear system of equations. The Newton solver, in turn, uses an arbitrary COMSOL Multiphysics linear solver for the resulting linear systems. Alternatively, you can use the COMSOL Multiphysics Newton solver to solve the nonlinear system. This gives you more 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. When you select the Generalized-α time-stepping algorithm, COMSOL Multiphysics uses the Newton solver. 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.

See Also

- The Nonlinear Solver Algorithm

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 they 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 example, all physics interfaces for transport problems—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 extra accuracy of the high-order BDF methods.

The Eigenvalue Solver Algorithm

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 Multiphysics 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. 6). 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.

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 Multiphysics determines the size of this step on the basis of the convergence speed of the Newton iteration using step-size selection criteria based on work in Ref. 7.

The Stationary Segregated Solver Algorithm

VERGENCE 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. Furthermore,

$$\mathrm{e}^N_{j,k} = \max_l(1-\alpha_l)\left[\frac{1}{N_j}\sum_{i=1}^{N_j}\left(\frac{\left|(\Delta U^{l,j,k})_i\right|}{W^j{}_i}\right)^2\right]^{1/2}$$

is an estimate of the largest damped Newton error. Here l is taken for all iterations in all substeps solving for the group j, α_l is the damping factor, $\Delta U^{l,j,k}$ is the Newton increment vector, and N_j is the number of DOFs. The weight factor $W^j{}_i$ is described below. Moreover,

$$\mathrm{e}^S_{j,k} = \left[\frac{1}{N_j}\sum_{i=1}^{N_j}\left(\frac{\left|(U^{j,k} - U^{j,k-1})_i\right|}{W^j{}_i}\right)^2\right]^{1/2}$$

is the relative increment over one complete iteration k. In this expression, $U^{j,k}$ is the segregated solution vector for the group j, and $W^j{}_i = \max(|U^j{}_i|, S_i)$, where S_i is a scale factor that the solver determines from the settings in the **Scaling** section of the Settings window for the **Variables** node.

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**, $W_i = 1$.

The Adaptive Solver Algorithm

THE L2 NORM ESTIMATE

The L2 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 L2 norm of the error, e_l, in the lth equation satisfies

$$\|e_l\| \le C\left\|h^{q_l}\rho_l\right\|$$

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 L2-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 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. 8). 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 basis 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 basis 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 the mesh elements.

The adaptive solver performs the following iterative algorithm (Ref. 9):

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 just 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 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 automatic time interval control a measure of 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, and we have 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 a 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.

The Sensitivity Solver Algorithm

When you enable sensitivity analysis, the stationary solvers compute—in addition to the basic forward solution—the sensitivity of a functional

$$Q = Q(u_p, p) \tag{9-1}$$

with respect to the sensitivity variables p. The forward solution u_p is a solution to the parameterized discrete forward problem

$$L(u_p, p) = N_F \Lambda_p \qquad M(u_p, p) = 0 \tag{9-2}$$

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:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \frac{\partial Q}{\partial u}\frac{\partial u}{\partial p} \tag{9-3}$$

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, Equation 9-2, 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

$$J\begin{pmatrix} \frac{\partial u_p}{\partial p} \\ \frac{\partial \Lambda_p}{\partial p} \end{pmatrix} = \begin{pmatrix} \frac{\partial L}{\partial p} \\ \frac{\partial M}{\partial p} \end{pmatrix} \qquad J = \begin{bmatrix} K & N_F \\ N & 0 \end{bmatrix} \tag{9-4}$$

solve for the sensitivities $\partial u_p/\partial p$ and $\partial \Lambda_p/\partial p$, and plug them back into Equation 9-3:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \begin{pmatrix} \frac{\partial Q}{\partial u} \\ 0 \end{pmatrix}^T J^{-1}\begin{pmatrix} \frac{\partial L}{\partial p} \\ \frac{\partial M}{\partial p} \end{pmatrix} \tag{9-5}$$

This formula gives dQ/dp explicitly in terms of known quantities, but in practice, it is too expensive to actually invert the matrix J.

If the number of individual sensitivity variables, p_j, is small, Equation 9-4 can be solved for each right-hand side $[\partial L/\partial p_j\ \partial \mathrm{M}/\partial p_j]^T$ and the solution inserted into Equation 9-3. 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 9-5:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \begin{pmatrix} u* \\ \Lambda* \end{pmatrix}^T \begin{pmatrix} \dfrac{\partial L}{\partial p} \\ \dfrac{\partial M}{\partial p} \end{pmatrix}$$

$$J^T \begin{pmatrix} u* \\ \Lambda* \end{pmatrix} = \begin{pmatrix} \dfrac{\partial Q}{\partial u} \\ 0 \end{pmatrix}$$

On 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. Obviously, this is much faster than the forward method if the number of variables is large. The system matrix, which must be factorized, is the transpose of the last linearization of the forward problem. If J is symmetric or Hermitian, this makes no difference compared to the forward method and all direct solvers can reuse the factorization. In the general case, however, MUMPS is more efficient than the others since it can reuse the factorization of J to perform back-substitution on $J^{-\mathrm{T}}$.

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

$$E\ddot{u} + D\dot{u} + Ku = L\,, \tag{9-6}$$

where E is the mass matrix, where 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 all constant. Either E or D can be identically equal to zero. When performing frequency response studies, the vector L is assumed to be harmonic. Before solving Equation 9-6, 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 as

$$u_m = \Phi_r q(t), \tag{9-7}$$

where $q(t)$ is a vector of unknown coefficients. Replacing u in Equation 9-6 by $\Phi_r q(t)$ and premultiplying by $\overline{\Phi_l}^T$ results in

$$\overline{\Phi_l}^T E\Phi_r\ddot{q} + \overline{\Phi_l}^T D\Phi_r\dot{q} + \overline{\Phi_l}^T K\Phi_r q = \overline{\Phi_l}^T L. \tag{9-8}$$

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 $\overline{\Phi_l}^T D\Phi_r$ as $\mathrm{diag}(2\xi_i\omega_i)$.

FREQUENCY RESPONSE

For a frequency response study the load L is assumed to be of the type

$$L(\Omega, t) = \tilde{L}(\Omega)e^{i\Omega t}, \tag{9-9}$$

where Ω is the frequency of the forcing function. The steady-state solution of Equation 9-8 is then of the form

$$q(t) = ce^{i\Omega t}. \tag{9-10}$$

Use the expression of L from Equation 9-9 and the expression of q from Equation 9-10 in Equation 9-8 to get

$$[-\Omega^2\overline{\Phi_l}^T E\Phi_r + i\Omega\overline{\Phi_l}^T D\Phi_r + \overline{\Phi_l}^T K\Phi_r]c = \overline{\Phi_l}^T \tilde{L}(\Omega). \tag{9-11}$$

An approximative solution to the original problem is given by $u_m = \Phi_r c$. This assumes that no non-homogeneous Dirichlet boundary conditions are present.

References for Solver Algorithms

1. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. The ARPACK Arnoldi package, www.caam.rice.edu/software/ARPACK.

7. P. Deuflhard, “A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques,” *Numer. Math.*, vol. 33, pp. 115–146, 1979.

8. 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.

9. R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Teubner Verlag and J. Wiley, Stuttgart, 1996.

10. www.netlib.org/ode.

Linear System Solvers

In this section:

The MUMPS Direct Solver

The parallel sparse direct linear solver MUMPS (MUltifrontal Massively Parallel sparse direct Solver) works on general systems of the form $Ax = b$. MUMPS 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.

MUMPS also includes out-of-core capabilities. 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 the *COMSOL Multiphysics Installation and Operations Guide* for further details. The LU factors are stored as blocks on the hard drive. The code is written in F90. For further details about MUMPS, see Ref. 1.

The SPOOLES Direct 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 the 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 Multiphysics uses SPOOLES version 2.2 developed by Cleve Ashcraft and collaborators (Ref. 2).

The PARDISO Direct Solver

The parallel sparse direct linear solver PARDISO 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 Multiphysics uses the PARDISO version developed by Olaf Schenk and collaborators (Ref. 3), which is included with Intel MKL (Intel Math Kernel Library).

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 $\mathrm{sign}(l_{ii})\varepsilon|PP_{\mathrm{MPS}}D_rAD_cP|_\infty$. You can specify the pivot threshold ε. The perturbation strategy is not as robust as ordinary pivoting. In order to improve the solution PARDISO uses iterative refinements.

PARDISO also includes out-of-core capabilities. The PARDISO out-of-core solver stores the LU factors on the hard drive. This minimizes the internal memory usage. The price 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 the *COMSOL Multiphysics Installation and Operations Guide* for further details. The LU factors are stored as blocks on the hard drive.

The GMRES Iterative Solver

This linear system solver uses the restarted GMRES (generalized minimum residual) method (see Ref. 4 and Ref. 5). 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*.

The value in the **Number of iterations before restart** field in the **General** section of the **Settings** window for the **Iterative** node specifies the number of iterations 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 is often very large to produce a preconditioner of such a high quality that the termination criteria are fulfilled for a small number of iterations and for a small restart value. 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.

Two slightly different versions of GMRES are available in COMSOL Multiphysics. The difference between these two versions is whether left or right preconditioning is used. Select the preconditioning type from the **Preconditioning** list. The default choice is left preconditioning. Normally, the two versions of GMRES have similar convergence behavior (see Ref. 6). If the preconditioner is ill-conditioned there could, however, be differences in the behavior.

Note

For information about the convergence criterion used by GMRES and the **Relative tolerance** and **Factor in error estimate** fields in the **Error** section, see Convergence Criteria.

If the solver does not converge, it terminates with an error message when it reaches the value in the **Maximum number of iterations** field (default = 10,000).

The FGMRES Iterative Solver

This solver uses the restarted FGMRES (flexible generalized minimum residual) method (see Ref. 7). 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 value in the **Number of iterations before restart** field. 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, for example, the Incomplete LU preconditioner, then the FGMRES solver is identical to the right preconditioned GMRES solver.

Note

For information about the convergence criterion used by FGMRES and the **Relative tolerance** and **Factor in error estimate** fields, see Convergence Criteria.

The Conjugate Gradients Iterative Solver

This solver uses the conjugate gradients iterative method (see Ref. 4, Ref. 8, and Ref. 9). 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.

Select the preconditioning type from the **Preconditioning** list. The default choice is left preconditioning. For the Conjugate gradient method this choice only affects the convergence criterion and not the algorithm itself.

Note

For information about the convergence criterion and the **Relative tolerance** and **Factor in error estimate** fields, see Convergence Criteria.

The BiCGStab Iterative Solver

This solver uses the biconjugate gradient stabilized iterative method (see Ref. 4 and Ref. 10) 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 requirement does not increase with the number of iterations as it does 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 may even be orders of magnitude larger than the initial residual, which can affect the numerical accuracy as well as the rate of convergence. The iterations are restarted with the current solution as initial guess if the algorithm detects poor accuracy in the residual or the risk for stagnation.

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 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.

Select the preconditioning type from the **Preconditioning** list. The default choice is left preconditioning.

Note

For information about the convergence criterion and the **Relative tolerance** and **Factor in error estimate** fields, see Convergence Criteria.

Convergence Criteria

When you use an iterative solver COMSOL Multiphysics estimates the error of the solution while solving. Once the error estimate is small enough, as determined by the convergence criterion

$$\rho\left|M^{-1}(b-Ax)\right| < \mathrm{tol}\cdot\left|M^{-1}b\right| \qquad (9\text{-}12)$$

the software terminates the computations and returns a solution. When you use a direct solver COMSOL Multiphysics can optionally make a check (**Error 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 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 Equation 9-12 states that the iterations terminate when the relative (preconditioned) residual times the factor ρ is less than a tolerance tol. You can set the factor ρ in the **Factor in error estimate** field (default = 400). 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: $|x - A^{-1}b| < \mathrm{tol} \cdot |A^{-1}b|$.

References for the Linear System Solvers

1. http://graal.ens-lyon.fr/MUMPS/

2. www.netlib.org/linalg/spooles/

3. www.pardiso-project.org/

4. Greenbaum, A., “Iterative Methods for Linear Systems,” *Frontiers in Applied Mathematics*, vol. 17, SIAM, 1997.

5. 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.

6. Y. Saad, *Iterative Methods for Sparse Linear Systems*, Boston, 1996.

7. Y. Saad, “A Flexible Inner-Outer Preconditioned GMRES Algorithm,” *SIAM J. Sci. Statist. Comput.*, vol. 14, pp. 461–469, 1993.

8. 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.

9. C. Lanczos, “Solutions of Linear Equations by Minimized Iterations,” *J. Res. Nat. Bur. Standards*, vol. 49, pp. 33–53, 1952.

10. 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.

Preconditioners for Iterative Solvers

In this section:

- The Incomplete LU Preconditioner
- The Geometric Multigrid Solver/Preconditioner
- The Algebraic Multigrid Solver/Preconditioner
- The SSOR, SOR, SORU, and Diagonal Scaling (Jacobi) Algorithms
- The SSOR Vector, SOR Vector, and SORU Vector Algorithms
- The SSOR Gauge, SOR Gauge, and SORU Gauge Algorithms
- The SOR Line Algorithm
- The Vanka Algorithm
- References for the Preconditioners for Iterative Solvers

The Incomplete LU Preconditioner

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. 1 and Ref. 2). 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.

In the **Drop using** list in the **Incomplete LU** node, you can select a drop rule (see the following section). Depending on the selected drop rule, you can specify a **Drop tolerance** or a **Fill ratio**. 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 on the other hand memory requirements and preconditioning time increases. 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.

You can also set the **Pivot threshold**, a number between 0 and 1 (default = 1). 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.

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. When using incomplete LU factorization as a preconditioner or smoother, it performs a fixed number of sweeps as dictated by the value in the **Number of iterations** field (default = 1).

Specify ω in the **Relaxation factor** field (default = 1).

See Also

- The SSOR, SOR, SORU, and Diagonal Scaling (Jacobi) Algorithms

SELECTING A DROP RULE

The Incomplete LU preconditioner uses either 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. The default is to keep these nonzeros. 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, crucial choice.

The Geometric Multigrid Solver/Preconditioner

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. 3 for more information.

COMSOL Multiphysics uses a hierarchy of *multigrid levels* where each 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 either “manually” or automatically. The automatic version applies 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 wish to control details in the meshes.

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 $A_1, \dots, A_N$ 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_ix 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 chose an arbitrary 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, the 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, they are 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.

See Also

- The SSOR, SOR, SORU, and Diagonal Scaling (Jacobi) Algorithms

NOTES ON THE EFFICIENCY OF SMOOTHERS

COMSOL Multiphysics performs smoothing on all but the coarsest multigrid level. A smoother should be computationally cheap and be effective at reducing the part of the error that has a high spatial frequency on the mesh to which it is applied. Therefore, the 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 type of problems, you should ensure that this criterion is fulfilled on the coarsest mesh if simple iterations is used as a smoother. In situations where the criterion is not fulfilled on coarse meshes GMRES can be a suitable smoother (Ref. 6).

The Algebraic Multigrid Solver/Preconditioner

The algebraic multigrid solver 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.

You can control the automatic construction of the multigrid hierarchy with the **Number of multigrid levels**, **Maximum number of DOFs at coarsest level**, and **Quality of multigrid hierarchy** fields in the **Multigrid** node. Coarse levels are added until the number of DOFs at the coarsest level is less than the max DOFs at coarsest level (default = 5000) or until it has reached the number of multigrid levels (default = 5). In the **Quality of multigrid hierarchy** field specify an integer value between 1 and 10 (default = 3) to make a trade-off between memory requirements and preconditioner quality. 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. In contrast, if the algebraic multigrid algorithm runs into memory problems, try a lower value to use less memory.

The remaining settings for the algebraic multigrid solver/preconditioner and its smoothers and coarse solver are identical to those for the geometric multigrid (GMG) solver.

See Also

- The Geometric Multigrid Solver/Preconditioner

The SSOR, SOR, SORU, and Diagonal Scaling (Jacobi) Algorithms

These simple and memory-efficient solvers/preconditioners/smoothers are 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.

The SOR (successive over-relaxation) method uses the same formula with $M = L + D/\omega$, where 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 A is symmetric, the SSOR method has the advantage that 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.

By default a blocked version of the SOR algorithms is used. It is optimized for parallel computations. In this case M is constructed from a column permuted version of A.

When these algorithms run as linear system solvers (by selecting **Use preconditioner** in the **Solver** list in the **Iterative** node), they perform sweeps until they have established convergence or they have reached the maximal number of iteration. You control this aspect with the parameters in the **Factor in error estimate**, and **Maximum number of iterations** fields in the **Iterative** node just like for any other iterative solver.

When the algorithms run as preconditioners or smoothers, they perform a fixed number of sweeps as dictated by the value in the **Number of iterations** field in the preconditioner node (default = 2). Here, you also specify ω in the **Relaxation factor (omega)** field (default = 1).

The SSOR Vector, SOR Vector, and SORU Vector Algorithms

These preconditioners/smoothers are intended for problems involving the $\nabla \times (a\nabla \times .)$ curl-curl operator and where you use *vector elements*. The vector elements are available primarily for electromagnetic wave simulations in the RF Module. The algorithm is an implementation of the concepts in Ref. 9 and Ref. 4. 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 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 (for clarity, boundary constraints are disregarded)

$$-\nabla \cdot (c\nabla\phi) = -\nabla \cdot \mathbf{F} \,,$$

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 does 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 non-vector shape functions are disregarded, and therefore you can 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 therefore the same as when you apply the standard SOR algorithm (see above).

The parameter in the **Number of iterations** field controls the number of main iterations (default = 2). For each main iteration, the algorithm makes a number of SOR iterations for the projected equation system; set that number (default = 1) in the **Number of secondary iterations** field.

In more detail, to preserve symmetry as a preconditioner and also when used as symmetric pre- and postsmoother in a multigrid setting, the SOR iterations are done in the following order:

- In each main iteration, the SOR vector version of this algorithm makes one SOR iteration on the main system followed by a number of secondary SOR iterations on the projected system.
- In each main iteration, the SORU vector version first makes a number of secondary SORU iterations on the projected system followed by one SORU iteration on the main system.
- In each main iteration, the SSOR vector version makes one SOR iteration on the main system followed by a number of secondary SSOR iterations on the projected system and then one SORU iteration on the main system.

You specify the relaxation factor ω in the **Relaxation factor** field (default = 1). It applies to all the different types of SOR iterations in this algorithm.

The SSOR Gauge, SOR Gauge, and SORU Gauge Algorithms

These preconditioners/smoothers are primarily intended for 3D magnetostatic problems in the AC/DC Module discretized with vector elements. The smoothers are basically SOR smoothers with some added functionality.

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, provided that 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; see "The SSOR Vector, SOR Vector, and SORU Vector Algorithms" on page 451. To this end, the system $T^T T\psi = -T^T b$ is first solved. Adding $T\psi$ to b then makes the numerical divergence of the right side small.

As in the case of SOR the blocked version is used by default. It performs better when running on a parallel machine.

In addition to the initial divergence cleaning, SOR gauge also performs a number of cleaning iterations in each linear solver iteration. You can control the number of such divergence cleaning iterations in the **Number of secondary iterations** field. The default number of secondary iterations is 1. In the **Variables** area you can specify which vector degrees of freedom to include in the divergence cleaning (this applies both to the initial and secondary cleaning iterations). By default, all vector degrees of freedom are included in the divergence cleaning.

The settings **Number of iterations** and **Relaxation factor** work in the same way as for the usual SOR smoothers; see "The SSOR, SOR, SORU, and Diagonal Scaling (Jacobi) Algorithms" on page 450.

The SOR Line Algorithm

In regions where the mesh is sufficiently anisotropic, the algorithm forms *lines* of nodes that connect nodes that are relatively close to each other (Ref. 12). 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 (like in the Vanka algorithm; 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.

See Also

- The Vanka Algorithm

SOR LINE SMOOTHER PARAMETERS

The **Number of iterations** field controls the number of smoothing iterations that are performed each time the SOR line smoother is invoked (default: 2).

The parameters in the **Main** section control the line updates. If **Method** is **Uncoupled** (default), each block SOR smoothing step updates a set of degrees of freedom with the same name that are located on a line. If **Method** is **Coupled**, 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 **Relaxation factor** field controls the damping of the block SOR smoothing steps (default is no damping). If your mesh elements have very large aspect ratios, the convergence can be improved if you decrease the relaxation factor to about 0.8. The **Blocked version** check box is selected by default, which means that the updates are performed in an order that is suited for parallel computations.

The parameters in the **Secondary** section control the SSOR updates. The value in the **Number of secondary iterations** field determines the number of SSOR iterations per SOR line iteration (default: 1). The **Relaxation factor** field controls the damping of the SSOR updates (default is no damping). If your mesh elements have very large aspect ratios, the convergence can be improved if you decrease the relaxation factor to about 0.8.

The Vanka Algorithm

This preconditioner/smoother is intended for, but not restricted to, problems involving the Navier-Stokes equations. Formally it applies to saddle-point problems. A saddle-point problem is a problem where the (equilibrium) solution is neither a maximum nor a minimum. The corresponding linear system matrix is indefinite, and often it has zeros on the diagonal. This is the case for the Navier-Stokes equations but also for problems formulated with weak constraints.

The algorithm is a local smoother/preconditioner of Vanka type. It is based on the ideas in Ref. 5, Ref. 11, and Ref. 12. 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 the Lagrange multiplier variable (or set of 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 defines the DOFs U_k, and the software 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 either with a LAPACK direct solver subroutine call or by a GMRES iterative method subroutine call. The GMRES method is the restarted GMRES without preconditioning. The algorithm relies on 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.

Note

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, you should use the **Direct** 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 since 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.

In general, the Vanka update does not necessarily update all DOFs. This is, for example, 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 (or similar types of equations) coupled to other equations, but where the coupling is not directly through the pressure variable. This is, for example, the case with the k-ε turbulence model. 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 denoted the *SSOR update*. The SSOR update 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.

A blocked version that works on a permuted version of the system matrix is used by default, as in the case of SOR. It is especially suited for parallel computations.

Control the number of Vanka updates and consecutive SSOR updates by the parameter in the **Number of iterations** field. For the Vanka update, control the Lagrange variables used for the local block definitions by the **Variables** area, and control the type of solver used for the block inverse operation by the **Block solver** list. If you choose GMRES, then you can control the convergence tolerance and the number of iterations before restart by the parameters in the **Tolerance** and **Number of iterations before restart** fields, respectively. Control the Vanka update relaxation parameter ω by the parameter in the **Relaxation factor** field. For the SSOR update, control the number of SSOR sweeps by the parameter in the **Number of secondary iterations** field and control the SSOR relaxation factor, used in these sweeps, with the parameter in the **Relaxation factor** field.

References for the Preconditioners for Iterative Solvers

1. J.R. Gilbert and S. Toledo, “An Assessment of Incomplete-LU Preconditioners for Nonsymmetric Linear Systems,” *Informatic*a, vol. 24, pp. 409–425, 2000.

2. 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.

3. W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

4. 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.

5. 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.

6. 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.

COMSOL’s implementations of the algebraic multigrid solver and preconditioner are based on the following references:

7. K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD Report 70, GMD, 1999.

8. C. Wagner, *Introduction to Algebraic Multigrid*, course notes, University of Heidelberg, 1999.

9. R. Hiptmair, “Multigrid Method for Maxwell’s Equations,” *SIAM J. Numer. Anal.*, vol. 36, pp. 204–225, 1999.

10. 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.

11. 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.

12. 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.

10

The COMSOL File Formats

This chapter describes the COMSOL file formats including the COMSOL Multiphysics files in binary format and text format.

In this chapter:

File Formats

The following table shows the file types that COMSOL Multiphysics can read and write and provides pointers to documentation:

TABLE 10-1: FILE FORMATS SUMMARY

FILE FORMAT	EXTENSION	PRODUCT	READ	WRITE	DOCUMENTATION
COMSOL Model File	mph	MPH	Yes	Yes	N/A
Model Java File	java	MPH	No	Yes	COMSOL Java API Reference Guide
Model Class File	class	MPH	Yes	No	N/A
Model M-File	m	LLML	No	Yes	COMSOL Java API Reference Guide
Binary Data File	mphbin	MPH	Yes	Yes	Binary Data Files and Text Data Files
Text Data File	mphtxt	MPH	Yes	Yes	Binary Data Files and Text Data Files
Spreadsheet file	txt	MPH	Yes	Yes	Spreadsheet Data Format
Grid file	txt	MPH	Yes	Yes	Grid Data Format
Sectionwise file	txt	MPH	Yes	Yes	Sectionwise Data Format
Continuous color tables		MPH	Yes	No	Continuous Color Tables
Discrete color tables		MPH	Yes	No	Discrete Color Tables

Product keys:

- MPH: COMSOL Multiphysics
- LLML: COMSOL LiveLink for MATLAB

Data Formats

COMSOL's data formats are used for exporting results data to file as well as representing input to interpolation functions in COMSOL Multiphysics. For all data formats, the exported file can contain a number of header rows starting with %, which contain information about the model and the exported data.

In this section:

- Spreadsheet Data Format
- Grid Data Format
- Sectionwise Data Format

Spreadsheet Data Format

This data format is used for importing unstructured data and exporting results data:

TABLE 10-2: DATA

SECTION	DESCRIPTION
Data	Data values separated by spaces

Each row of the file defines the coordinates and data values for the function in one point. A file used to define a function of three variables can begin as follows:

```
0 0 0.12 0.34
0 1 0.52 1.50
1 0 0.67 0.91
...
```

The first variable (input argument) appears in the first (leftmost) column, the second variable in the second column, the third variable in the third column, and the function values in the fourth (rightmost) column. The variables can be any function inputs. For space-dependent functions $f(x, y, z)$ they are the x-, y-, and z-coordinates.

It is possible to define several functions in one file by providing more than one data column after the input variables (which, in many cases, are coordinates). When exporting several expressions to a file, COMSOL Multiphysics generates files with this structure.

Grid Data Format

Note

The grid data format can only be used for import of data.

The following table shows the format for results data stored as grid points and corresponding data values:

TABLE 10-3: GRID, DATA

SECTION	NO. ROWS	DESCRIPTION
% Grid	1–3	x grid points separated by spaces y grid points separated by spaces (optional) z grid points separated by spaces (optional)
% Data	3 (2D), 4 (3D)	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.

Important

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.

It is possible to include more than one function in the file as long as a `% Data` header separates them one from the other.

Sectionwise Data Format

The following table shows the format for results data stored as node coordinates, elements, and corresponding data values:

TABLE 10-4: NODES, ELEMENTS, DATA

SECTION	NO. COLUMNS	DESCRIPTION
`%Coordinates`	1–3	One to three columns containing x, y (optional), and z (optional)
`%Elements`	3 (2D), 4 (3D)	Triangulation where each row contains the row indices of the points in the Coordinates section that make up a single element—triangular in 2D, tetrahedral in 3D
`%Data`	1	Column of data values

This format can also be used to import data for unstructured interpolation. It has the advantage over the Spreadsheet Data Format in that it also contains the exact mesh used to perform the interpolation.

Color Tables

Color table files are used when plotting. When COMSOL Multiphysics is loaded, it reads all files in the following directories:

- `data/colortables/` in the directory where COMSOL Multiphysics is installed.
- The user settings directory `.comsol/v42` under your local home directory.

The files that adhere to the format specified in this section are made available as color tables. In this section:

- Continuous Color Tables
- Discrete Color Tables

Continuous Color Tables

An example of a color table that defines a transition from blue to red:

```
% Continuous
0 0 1
1 0 0
```

Each line, from left to right, contains the red, green, and blue components (RGB) of a color. The components are floating-point values between 0 and 1. The table can contain an arbitrary number of colors.

Each color can contain an optional fourth component, the length of the interval between the two colors. For N colors, there are N–1 intervals. An example of a color scale that defines a sharp transition from blue to magenta followed by a slow transition from magenta to red:

```
% Continuous
0 0 1 1
0.5 0 0.5 10
1 0 0
```

Discrete Color Tables

An example of a color table for which the lower half of the legend is blue and the upper half is red:

```
% Discrete
0 0 1
```

```
1 0 0
```

Each color can contain an optional fourth component, the length of the interval occupied by that color. For *N* colors, there are *N* intervals. An example of a color scale that defines a small blue interval, a long green interval, and a medium blue interval:

```
% Discrete
0 0 1 1
0 1 0 10
1 0 0 5
```

Binary Data Files and Text Data Files

A COMSOL Multiphysics file is used to store COMSOL Multiphysics data. The format is suitable for exchange of mesh or CAD data between COMSOL Multiphysics and other software systems. It is possible to save a COMSOL Multiphysics file in a text file format, using the extension .mphtxt, or a binary file format, using the extension .mphbin. The file formats contain the same data in the same order.

In this section:

- File Structure
- Records
- Terminology
- Text File Format
- Binary File Format

File Structure

The COMSOL Multiphysics file format has a global version number, so that it is possible to revise the whole structure. The first entry in each file is the file format, indicated by two integers. The first integer is the major file version and the second is referred to as the minor file version. For the current version, the first two entries in a file is `0 1`.

The following sections describe the file structure of the supported version.

FILE VERSION 0.1

After the file version, the file contains three groups of data:

- A number of *tags* stored as strings, which gives an identification for each *record* stored in the file.
- A number of *types*, which are strings that can be used in serializing the object. The tag should be used as an indication of the contents of the file, but can also be an empty string.
- The *records* containing the serialized data in the file.

Example When using `model.mesh(<tag>).export(<filename>)` or **Mesh>Export to File** to save a COMSOL Multiphysics mesh, the tag equals the variable name (m1), the type is set to obj (but this is not used), and the record contains the serialization of the mesh object, including point coordinates and element data of the mesh. See Examples for more examples of COMSOL Multiphysics text files.

```
# Created by COMSOL Multiphysics Fri Aug 26 14:19:54 2005

# Major & minor version
0 1
######### Tags
1 # number of tags
2 m1

######## Types
1 # number of types
3 obj

######## Records

# A planar face object

0 0 1
4 Mesh # class
...
```

Records

The record contains the serialization data in the file and additional information on how to process the serialized data. It also has a version number.

The record is a wrapper for *serializable types* stored in the file. The reason for having this wrapper is to be able to use a version number, so that the serialization can be revised in the future while maintaining backward compatibility.

The following sections describe the format of the supported version:

RECORD VERSION 0

This record is a wrapper for *serializable types* stored in the file. The following table contains the attributes of the records:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
Integer		Version
Integer		Not used
Integer	type	Serialization type, 1 for Serializable
Serializable	obj	If type equals 1, this field follows

Serialization type 1 indicates that the following field is a subtype to `Serializable`. COMSOL Multiphysics uses type equal to 0 internally, but such files are only used for temporary purposes.

Terminology

The following data types are used in the serialization:

- *Boolean* refers to an 8-bit signed character which must be 0 or 1.
- *Character* refers to an 8-bit signed character.
- *Integer* refers to a 32-bit signed integer.
- *Double* refers to a 64-bit double.

Matrices are stored in row-major order. In this documentation brackets are used to indicate a matrix. Hence, `integer[3][4]` means that 12 integers representing a matrix are store in the file. The first three entries corresponds to the integers in the first row of the matrix, and so on.

Text File Format

COMSOL Multiphysics text file, using the file extension .mphtxt, are text files where values are stored as text separated by whitespace characters.

Lexical conventions:

- Strings are serialized as the length of the string followed by a space and then the characters of the string, for example, "`6 COMSOL`". This is the only place where whitespace matters.
- The software ignores everything following a # on a line except when reading a string. This makes it possible to store comments in the file.

Binary File Format

COMSOL Multiphysics binary file, using the extension .mphbin, are binary files with the following data representation:

- Integers and doubles are stored in little-endian byte order.
- Strings are stored as the length of the string (integer) followed by the characters of the string (integers).

Serializable Types

Attribute

BezierCurve

BezierMfd

BezierSurf

BezierTri

BSplineCurve

BSplineMfd

BSplineSurf

Ellipse

Geom0

Geom1

Geom2

Geom3

Manifold

Mesh

MeshCurve

MeshSurf

Plane

PolChain

Serializable

Solution

Straight

Transform

VectorDouble

VectorInt

VectorString

Attribute

Supported Versions 0

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	n	Number of attribute fields
Serializable[n]		Each entity field is stored as a serializable

Description An `Attribute` is a general purpose class from which different subtypes can be derived. Each such subtype should then be serialized using the serialization of the `Attribute` class, which means that all that it should add to the serialization is the version number.

Attributes are used in COMSOL Multiphysics for internal purposes, and these attributes are not documented. However, because `Attribute` has a strict serialization structure, the serialization of these attributes is well documented.

Example This is a serialization of an attribute used internally in COMSOL Multiphysics. It is serialized as a vector of integers.

```
11 AssocAttrib # class
0 0 # version
1 # nof attribute fields
9 VectorInt # class
18 3 2 2 2 1 0 1 1 1 1 1 1 0 1 1 1 1 0
```

Supported Versions 0

Subtype of BezierMfd

Fields The class is defined by the following fields:

ENTITY/OBJECT	DESCRIPTION
integer	Version
BezierMfd	Parent class containing common data

Description A rational Bézier curve is a parameterized curve of the form

$$\mathbf{b}(t) = \frac{\sum_{i=0}^{p} \mathbf{b}_i w_i B_i^p(t)}{\sum_{i=0}^{p} w_i B_i^p(t)}, 0 \le t \le 1$$

where the functions

$$B_i^p(t) = \binom{p}{i} t^i (1-t)^{p-i}$$

are the *Bernstein basis* functions of *degree* p, $\mathbf{b}_i = (x_1, \ldots, x_n)$ are the control vertices of the n-dimensional space, and w_i are the weights, which should always be positive real numbers to get a properly defined rational Bézier curve. A rational Bézier curve has a direction defined by the parameter t.

Example The following illustrates a linear Bézier curve.

```
11 BezierCurve # class
0 0 # version
2 # sdim
0 2 1 # transformation
1 0 # degrees
2 # number of control points
# control point coords and weights
0 0 1
1 1 1
```

See also BSplineCurve

Supported Versions 0

Subtype of Manifold

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation class
integer	m	Degree in first parameter
integer	n	Degree in second parameter
integer	k	Number of control points
double[k][d+1]	P	Matrix of control points with the weights in the last column

Description The `BezierMfd` type is the abstract base class for the different type of Bézier surfaces and curves that are supported. These can all be represented in using the generalized equation

$$\mathbf{S}(s,t) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{b}_{i,j}w_{i,j}B(s,t)}{\sum_{i=0}^{m}\sum_{j=0}^{n}w_{i,j}B(s,t)}$$

where B are functions as described in the respective entry, and $\mathbf{b}$ are the control point coordinates in `P` and w are the weights stored in the last column of `P`.

See also BSplineMfd

Supported Versions 0

Subtype of BezierMfd

Fields The class is defined by the following fields:

ENTITY/ OBJECT	DESCRIPTION
integer	Version
BezierMfd	Parent class containing common data

Description A rectangular rational Bézier patch of degree p-by-q is described by

$$\mathbf{S}(s,t) = \frac{\sum_{i=0}^{p}\sum_{j=0}^{q}\mathbf{b}_{i,j}w_{i,j}B_i^p(s)B_j^q(t)}{\sum_{i=0}^{p}\sum_{j=0}^{q}w_{i,j}B_i^p(s)B_j^q(t)},\ 0 \le s, t \le 1\ ,$$

where B_i^p and B_j^q are the Bernstein basis functions of degree p and q, respectively, as described in the entry of BezierCurve. This surface description is called rectangular because the parameter domain is rectangular, that is, the two parameters s and t can vary freely in given intervals.

See also BSplineSurf, BezierTri

Supported Versions 0

Subtype of `BezierMfd`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	DESCRIPTION
integer	Version
BezierMfd	Parent class containing common data

Description Another form of surface description is the triangular patch, also called a *Bézier triangle*. A triangular rational Bézier patch is defined as

$$\mathbf{S}(s,t) = \frac{\sum_{i+j\le p} \mathbf{b}_{i,j} w_{i,j} B^p_{i,j}(s,t)}{\sum_{i+j\le p} w_{i,j} B^p_{i,j}(s,t)} \quad, 0 \le s, t \le 1$$

which differs from the Bézier curve description only by the use of *bivariate* Bernstein polynomials instead of *univariate*, for the curve case. The bivariate Bernstein polynomials of degree p are defined as

$$B^p_{i,j}(s,t) = \frac{p!}{i!j!(p-i-j)!} s^i t^j (1-s-t)^{p-i-j}, \quad i+j \le p$$

where the parameters s and t must fulfill the conditions

$$\begin{cases} 0 \le s, t \\ s + t \le 1 \end{cases}$$

which form a triangular domain in the parameter space, therefore the name of this surface description.

See also `BezierSurf`

Supported Versions 0

Subtype of BSplineMfd

Fields The class is defined by the following fields:

ENTITY/ OBJECT	DESCRIPTION
integer	Version
BSplineMfd	Parent class containing common data

Description The BSplineCurve, describes a general spline curve, using B-spline basis functions, as defined in BSplineMfd. Splines on this form are often referred to as B-splines.

A *p*th-degree spline curve is defined by

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_i^p(u) w_i \mathbf{P}_i}{\sum_{i=0}^{p} N_i^p(u) w_i}, a \le u \le b$$

where $\mathbf{P}_i$ are the control points., the wi are the weights and the N_i^p are the *p*th degree B-spline basis functions defined in the nonperiodic and nonuniform knot vector

$$U = \{a, \ldots, a, u_{p+1}, \ldots, u_{m-p-1}, b, \ldots, b\}$$

For non-rational B-splines, all weights are equal to 1 and the curve can be expressed as

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_i^p(u) w_i \mathbf{P}_i, a \le u \le b$$

See also BezierCurve

Supported Versions 0

Subtype of Manifold

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation class
integer		Dimension (1 if curve, 2 if surface)
integer	p, q	Degree in each dimension (1 or 2 integers)
boolean		If rational equal to1
integer		Number of knot vectors (1 for curves, 2 for surfaces)
integer	m1	Length of first knot vector
double[m1]	U	First knot vector
integer	m2	Length of second knot vector (not for curves)
double[m2]	V	Second knot vector (not for curves)
integer	n1	Number of control points in first parameter direction
integer	n2	Number of control points in second parameter dimension
integer	n3	Number of coordinates per control point
double [n1][n2][n3]	P	Matrix of coordinates where the last dimension is increased by 1 to store the weights if the manifold is rational

Description The `BSplineMfd` type is the abstract base type for `BSplineCurve` and `BSplineSurf`, which represent general spline curves and surfaces, respectively.

They are represented using *B-spline basis functions*. Let $U = \{u_0, \ldots, u_m\}$ be a nondecreasing sequence of real numbers; U is called the *knot vector* and the elements u_i of U are called *knots*. The ith B-spline basis function of p-degree, $N_i^p(u)$, is defined as

$$N_i^0(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^p(u) = \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u)$$

A general B-spline can be described by

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_i^p(u) N_j^q(v) w_{i,j} \mathbf{b}_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_i^p(u) N_j^q(v) w_{i,j}}, a \le u \le b, c \le v \le d$$

where

$$U = \{a, \ldots, a, u_{p+1}, \ldots, u_{m-p-1}, b, \ldots, b\}$$

and

$$V = \{c, \ldots, c, v_{p+1}, \ldots, v_{m-p-1}, d, \ldots, d\}$$

are the two knot vectors stored in the entry, and $\mathbf{b}$ and w are the control points coordinates and weights stored in P.

For periodic splines, the first and last parameter values in the knot vectors are not duplicated.

BSplineSurf

Supported Versions 0

Subtype of BSplineMfd

Fields The class is defined by the following fields:

ENTITY/ OBJECT	DESCRIPTION
integer	Version
BSplineMfd	Parent class containing common data

Description The generalization of B-spline curves to surfaces is a tensor product surfaces given by

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)N_j^q(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)N_j^q(v)w_{i,j}}, a \le u, v \le b$$

See also BezierSurf

Supported Versions 0

Subtype of `Manifold`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation class
double[d]	center	Center coordinate
boolean		Equal to 1 if clockwise rotation (only if d==2)
double[d]	normal	Normal vector coordinates,
double[d]	M	Major axis
double	rat	Ratio of minor axis length to major axis length
double	offset	Parameter at the end of major axis
boolean		Equal to 1 if ellipse is degenerated

Description This manifold defines an ellipse in the two or three dimensional space.

In 2D, an ellipse is defined by a center point `center`, a vector defining the major axis `M` of the ellipse (including the magnitude of the major axis), the radius ratio of the minor axis length to the major axis length `rat`, the direction of the ellipse, and the parameter offset at the major axis `offset`.

In 3D, an ellipse is defined by a center point `center`, a unit vector normal to the plane of the ellipse `normal`, a vector defining the major axis of the ellipse `M` (including the magnitude of the major axis), the radius ratio, and the parameter offset at the major axis `offset`. The direction of the ellipse is defined by the right hand rule using the normal vector.

An ellipse is a closed curve that has a period of 2π. It is parameterized as:

```
point = center + M cos(t - offset) + N sin(t - offset)
```

where `M` and `N` are the major and minor axes respectively.

Example

```
7 Ellipse # class
0 0 # version
2 # sdim
0 2 1 # transformation
0 0 # center
```

```
0 # reverse?
2 0 # major axis
0.5 # minor axis length / major axis length
0 # parameter value at end of major axis
0 # degenerated?
# Attributes
0 # nof attributes
```

Supported Versions 1

Subtype of Serializable

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	type	Geometry type
double		Relative geometry tolerance
integer	p	Number of points (0 or 1)
integer	na	Number of attributes
Attribute[na]		Vector of attributes

Description The type represent a 0D geometry class.

The type can be either 0 for solid or –1 for general object.

Example A solid 0D geometry object.

```
5 Geom0 # class
1 0 1e-010 1
0 # nof attributes
```

Geom1

Supported Versions 1, 2

Subtype of Serializable

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	type	Geometry type
integer		1 if void regions are labeled, 0 otherwise. Not present in version 1.
double		Geometry tolerance
integer	nv	Number of vertices
double[nv]	vtx	Vector of vertex coordinates
integer[nv][2]	ud	Matrix of integers giving domains on up and down side of each vertex

Description The type represent a 1D geometry class.

The type can be either 0, 1, or –1 for point, solid, or general objects, respectively.

Example A solid 1D object.

```
5 Geom1 # class
2 # version
1 # type
1 # voidsLabeled
1e-010 # gtol
3 # number of vertices
# Vertex coordinates
0
1
3
# Vertex up/down
1 0
2 1
0 2
# Attributes
0 # nof attributes
```

Supported Versions 1, 2

Subtype of Serializable

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer		Type
integer		1 if void regions are labeled, 0 otherwise. Not present in version 1.
double		Relative geometry tolerance
integer	nv	Number of vertices
integers/doubles [nv][4]	vertex	Matrix of vertex data
integer	ne	Number of edges
integers/doubles [ne][8]	edge	Matrix of edge data
integer	nc	Number of curves
Manifold[nc]	curve	An array of Manifold objects

Description The type represent a 2D geometry class.

The type can be either 0, 1, 2, or –1 for point, curve, solid or general objects, respectively.

Example

```
5 Geom2 # class
2 # version
1 # type
1 # voidsLabeled
1e-010 # gtol
0.0001 # resTol
2 # number of vertices
# Vertices
# X Y dom tol
0 0 -1 NAN
1 2.2999999999999998 -1 NAN

1 # number of edges
# Edges
# vtx1 vtx2 s1 s2 up down mfd tol
1 2 0 1 0 0 1 NAN
1 # number of manifolds
11 BezierCurve # class
```

```
0 0 # version
2 # sdim
0 2 1 # transformation
1 0 # degrees
2 # number of control points
0 0 1
1 2.2999999999999998 1
0 # nof attributes
```

Supported Versions 1, 2

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	type	Type
integer		1 if void regions are labeled, 0 otherwise. Not present in version 1.
double		Relative geometry tolerance
double		Relative resolution tolerance
integer	nv	Number of vertices
integers/doubles [nv][5]	vertex	Matrix of vertex data
integer	npv	Number of parameter vertices
integers/doubles [npv][6]	pvertex	Matrix of parameter vertex data
integer	ne	Number of edges
integers/doubles [ne][7]	edge	Matrix of edge data
integer	npe	Number of parameter edges
integers/doubles [nep][10]	pedge	Matrix of parameter edge data
integer	nf	Number of faces
integers/doubles [nf][4]	face	Matrix of face data
integer	nm	Number of 3D manifolds, curves and surfaces
Manifold[nmfd]	mfd	Vector of manifolds
integer	npc	Number of parameter curves
Manifold[npc]	pcurve	Vector of parameter curves

Description The type represent a 3D geometry class.

The type can be either 0, 1, 2, 3, or –1 for point, curve, shell, solid, or general objects.

Supported Versions 0

Subtype of `Serializable`

Fields This is an abstract class with no fields.

Description A manifold is the common supertype for curve and surface types. It is used by the geometry types.

See also `Geom0`, `Geom1`, `Geom2`, `Geom3`

Mesh

Supported Versions 1

Subtype of Serializable

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension (if equal to 0 no more fields)
integer	np	Number of mesh points
integer		Lowest mesh point index
double[d][np]	p	Mesh points
integer	nt	Number of element types (fives the number of repeats of the following fields)
string		Element type
integer	nep	Number of nodes per element
integer	ne	Number of elements
integer[ne][nep]	elem	Matrix of point indices for each element.
integer	ner	Number of parameter values per element
integer	nr	Number of parameter sets
double[nr][ner]	par	Matrix of parameter values
integer	ndom	Number of domain values
integer[ndom]	dom	Vector of domain labels for each element.
integer	nud	Number of up/down boundary relations
integer[nud]	ud	Matrix of integers stating domain number on upside and downside of the boundary

Description The domain numbering for points, edges, and boundaries must start from 0 when defining a mesh through a COMSOL Multiphysics mesh file.

Example The following displays a mesh with triangular elements on a unit square. Neither point or edge elements are present.

```
4 Mesh # class
2 # sdim
5 # number of mesh points
0 # lowest mesh point index
# Mesh point coordinates on unit square
0 0
1 0
1 1
```

```
0 1
0.5 0.5
1 # number of element types
3 tri # type name
3 # number of nodes per element
4 # number of elements
# Elements, 4 triangular elements
0 1 4
3 0 4
2 3 4
1 2 4
6 # number of parameter values per element
0 # number of parameters
4 # number of domains
# Domains
1
1
2
2
0 # number of up/down pairs
```

MeshCurve

Supported Versions 1

Subtype of `Manifold`

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer		Space dimension
Transform		Transformation
integer	np	Number of points
double[np][d]	p	Matrix of point coordinates
double[np]	par	Vector of point parameters
Manifold	intcurve	Interpolating curve

Description Mesh structures can also be used to define manifolds. Because meshes contain a number of nodes and, in the case of COMSOL Multiphysics, corresponding parameter values, a good geometric representation can be obtained using a suitable interpolation method for evaluating the values of the manifolds and its derivatives on parameter values that are inside the intervals between the given nodes. Mesh curves are handled by cubic spline interpolation.

The matrix `p` and the vector `par` corresponds to the structures corresponding structures in an edge mesh representation. For the MeshCurve, they serve as the interpolation data to obtain `intcurve`.

See also `BSplineCurve`

Supported Versions 1

Subtype of `Manifold`

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer		Space dimension
Transform		Transformation
integer	nv	Number of vertices
double[nv][3]	p	Matrix of mesh vertex coordinates
double[nv][2]		Matrix of mesh vertex parameters
integer	nt	Number of triangles
integers[nv][3]	elem	Matrix of vertex indices for each element

Description Mesh structures for surface meshes can be used to make a geometric definition of unstructured data. Since a mesh type in COMSOL Multiphysics have coordinates as well as parameter values for each element, interpolation can be employed to create smooth surfaces.

A quadratic interpolation is used to define a parametric surface from a surface mesh.

The matrix `p` of coordinates and the triangles `elem` with indices into p are used as the interpolation data.

See also `Mesh`

Plane

Supported Versions 1

Subtype of `Manifold`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation
double[d]	p	The point in the plane with parameter value (0,0)
double[d]	n	Normal vector
double[d]	b	Direction of first parameter axis

Description This manifold defines a plane in the three dimensional space. It is represented by a point, a unit vector normal to the plane, and the vector of the u derivative.

A plane is open in both parameter directions and neither periodic nor singular at any point. It is parameterized as:

```
pos = p + u*b + v*(n x b)
```

Example

```
5 Plane # class
0 0 # version
3 # sdim
0 3 1 # transformation
1.3 0.80000000000000004 1.6000000000000001 # root point
-6.1257422745431001e-017 0 1 # normal
-1 0 -6.1257422745431001e-017 # derivatives
0 # degenerated?
```

Supported Versions 1

Subtype of `Manifold`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation
integer	n1	First dimension of matrix of point coordinates (equal to d)
integer	n2	Second dimension of matrix point coordinates, number of points
doubles	pol	n1-by-n2 matrix of point coordinates

Description Polygon chains are piecewise linear curves that are used as approximations of curves in the decomposition algorithm. They have an implicit parameter representation; that is, $[(i-1)(p-1), i(p-1)]$ on the ith interval in a polygon chain with p points. This is not a suitable representation because the derivatives can vary substantially along the curve.

See also `MeshCurve`

Serializable

Supported Versions 0

Subtype of

Fields This is the abstract base type of all other types. It has no fields.

Description Serializable is the abstract base type. It is used to indicate that a field can contain all supported types, as is the case for the Attribute type.

See also `Attribute`

Supported Versions 1

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/OBJECT	VARIABLE	DESCRIPTION
integer		Solution object version
integer		Solution type
integer		Equal to one if it is ok to interpolate between solution vectors. Zero otherwise
string[]		Parameter names
integer		Number of degrees of freedom
integer		Mesh case number
integer		Size of parameter list
integer	nU	Number of data types in object
integer[nU]	iU	Number of solution vectors of each type
dcmplx[nU][iU]		Solution data identifier
integer[3]		DOF indices
Vector		Values of static DOFs
double		Parameter values
Vector		Values of dynamic DOFs without stored time derivatives
Vector		Values of dynamic DOFs with stored time derivatives
Vector		Time derivatives of corresponding DOFs

The first section of solution object specific data contains the following parts and in the following order:

- Internal solution object version number. At the time of writing the version number is 5.
- Solution type. The solution type is one of stationary solution(=0), parametric solution(=1), time dependent solution(=2), and eigenvalue solution(=3).
- Whether or not interpolation between solution vectors is ok.
- Parameter names. Text file specific information: The first integer gives the number of parameter names (a single empty parameter name also registers as a 1 here). Thereafter each parameter is stored preceded by its length.

- Total number of degrees of freedom. This number includes both variables solved for and variables not solved for.
- Mesh case number.
- Size of the parameter list. This also becomes the number of stored dynamic solution vectors described below (if any such components are present).

The second section of solution specific data stores information about the type of each DOF and the values of any static DOFs. The data is stored in the following order:

- Internal solution object version number. At the time of writing the version number is 5.
- Number of different data types used in the present version of the solution object. At the time of writing the data types are: ordinary solution data, reaction forces data, adjoint sensitivity data, forward sensitivity data, functional sensitivity data, and linearization point data (and they are stored in this order).
- Number of stored solutions of each type. This number specifies the number of solutions per parameter value and not the total number of solution vectors of a certain type. Text file specific information: Space-separated vector of integers with information about the number of stored solutions of each type (ordinary solution data, reaction forces data, and so on).
- Solution data identifiers. Text file specific information: Space-separated vector of complex numbers (stored as real part followed by a space and the imaginary part) containing solution data identifiers. First, the identifiers for all solutions of the first type is stored (where the number of solutions of the first type is given by the first integer of the preceding line). Then the identifiers for all solutions of the second type follows etc.
- Each DOF can be one of three types: static, dynamic without stored time derivatives, and dynamic with stored time derivatives. Information about which DOFs belong to which type is stored in three integer vectors. Text file specific information: The first vector contains the number of static DOFs followed by the indices of these DOFs. The second vector contains the number of dynamic DOFs stored without time derivatives followed by the indices of these DOFs. The third vector contains the number of dynamic DOFs stored with time derivatives followed by the indices of these DOFs.
- Vectors containing values of the static DOFs for one data type. Text file specific information: The first number gives the number of static vectors stored for the

current data type. Thereafter the actual vectors are stored preceded by their lengths.

One DOF index data structure and one static DOFs values data structure is needed for each data type. The DOF indices and static vectors sections are therefore repeated `nU` times.

The third section of solution specific data stores information about the dynamic part of the solution. In total this section contains as many subsections as the size of the parameter list. Each subsection contains the following information and in the following order:

- Internal solution object version number. At the time of writing the version number is 5.
- Number of different data types used in the present version of the solution object. At the time of writing the data types are: ordinary solution data, reaction forces data, adjoint sensitivity data, forward sensitivity data, functional sensitivity data, and linearization point data (and they are stored in this order).
- Parameter values for the data in the current subsection. The number of parameter values here is the same as the number of parameter names.
- Vector containing values of dynamic DOFs stored without time derivatives for one data type and the corresponding parameter values. Text file specific information: The first number gives the number of vectors stored for the current data type and DOF type. Thereafter the actual vectors are stored preceded by their lengths.
- Vector containing values of dynamic DOFs stored with time derivatives for one data type and the corresponding parameter values. Text file specific information: The first number gives the number of vectors stored for the current data type and DOF type. Thereafter the actual vectors are stored preceded by their lengths.
- Vector containing values of time derivatives for one data type and the corresponding parameter values. Text file specific information: The first number gives the number of vectors stored for the current data type and DOF type. Thereafter the actual vectors are stored preceded by their lengths.

Within each subsection dynamic DOF value data are stored for one data type at a time.

Example of text file output of a simple stationary solution object. The underlying setup is a simple one-dimensional geometry containing a single second order

Lagrange element with two uncoupled unknown variables. Both variables are solved for:

```
# Created by COMSOL Multiphysics Tue Jun 30 13:08:00 2009


# Major & minor version
0 1
1 # number of tags
# Tags
4 sol1
1 # number of types
# Types
3 obj

# --------- Object 0 ----------

0 0 1
8 Solution # class

5 # solution object version
0 # solution type
0 # interpolation between solnums is ok
1 0  # parameter names
6 # number of DOFs
0 # mesh case number
1 # size of parameter list

5 # solution object version
6 # number of data types
# number of solutions
1 1 0 0 0 0
# solution data identifier
NAN NAN NAN NAN
# DOF indices
6 0 1 2 3 4 5
0
0
# static vectors
1 6 1 1 1.375 1.0625 1 1
# DOF indices
4 0 1 4 5
0
0
# static vectors
1 4 -1.5 -0.500 -1.500 -0.500
# DOF indices
0
0
0
```

```
# static vectors
# DOF indices
0
0
0
# static vectors
# DOF indices
0
0
0
# static vectors
# DOF indices
0
0
0
# static vectors

5 # solution object version
6 # number of data types
1 1 0 # parameter values
# DOFs without time derivatives
1 0
# DOFs with time derivatives
1 0
# time derivatives
1 0
# DOFs without time derivatives
1 0
# DOFs with time derivatives
1 0
# time derivatives
1 0
```

The object only contains static solution data and that the only data types that are present are ordinary solution data and reaction forces data. The reaction forces data vector contains four entries since the setup has two unknown variables and both have been solved for in the example.

A second example uses the same setup as the previous one with the difference that a parametric solve with the parameters `p1` and `p2` has been performed. The parameter list was selected as `0 0 0.1 0.2` and only one of the two variables is solved for:

```
# Created by COMSOL Multiphysics Tue Jun 30 13:08:29 2009


# Major & minor version
0 1
1 # number of tags
```

```
# Tags
4 sol1
1 # number of types
# Types
3 obj

# --------- Object 0 ----------

0 0 1
8 Solution # class

5 # solution object version
1 # solution type
0 # interpolation between solnums is ok
2 2 p1 2 p2 # parameter names
6 # number of DOFs
0 # mesh case number
2 # size of parameter list

5 # solution object version
6 # number of data types
# number of solutions
1 1 0 0 0 0
# solution data identifier
NAN NAN NAN NAN
# DOF indices
3 1 3 5
3 0 2 4
0
# static vectors
1 3 0 0 0
# DOF indices
0
2 0 4
0
# static vectors
1 0
# DOF indices
0
0
0
# static vectors
# DOF indices
0
0
0
# static vectors
# DOF indices
0
0
```

```
0
# static vectors
# DOF indices
0
0
0
# static vectors

5 # solution object version
6 # number of data types
1 2 0 0 # parameter values
# DOFs without time derivatives
1 3 1 1.375 1
# DOFs with time derivatives
1 0
# time derivatives
1 0
# DOFs without time derivatives
1 2 -1.5 -1.500
# DOFs with time derivatives
1 0
# time derivatives
1 0

5 # solution object version
6 # number of data types
1 2 0.100 0.200 # parameter values
# DOFs without time derivatives
1 3 1 1.375 1
# DOFs with time derivatives
1 0
# time derivatives
1 0
# DOFs without time derivatives
1 2 -1.5 -1.500
# DOFs with time derivatives
1 0
# time derivatives
1 0
```

Half the DOFs end up in the static part and half in the dynamic part since only one variable is being solved for. This also has the consequence that the only appearing reaction force data in the stored solution object is in the dynamic part and that these vectors only have two components.

Straight

Supported Versions 1

Subtype of Manifold

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer		Version
integer	d	Space dimension
Transform		Transformation
double[d]	root	The point from which the ray starts
double[d]	dir	The direction in which the ray points
double	pscale	Parameter scale

Description This manifold defines an infinite straight line in the two or three dimensional space. It is represented by a point and a unit vector specifying the direction. A straight also has a scale factor for the parameterization, so the parameter values can be made invariant under transformation. If not specified the value of this parameter is set to 1.0.

A straight line is an open curve that is not periodic. It is parameterized as:

```
pos = root + u*pscale*dir
```

where u is the parameter.

Example

```
8 Straight # class
0 0 # version
3 # sdim
0 3 1 # transformation
1.3 0.8 0.0 # root point
-1 0 0 # direction
1 # parameter scale
```

See also Plane

Supported Versions 1

Subtype of `Serializable`

Fields The class is

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer	d	Space dimension
boolean		1 if transformation is a unit transformation, 0 otherwise. If the value is 1, no more fields are present
double [d+1][d+1]	M	Values in transformation matrix
boolean		1 if determinant is positive, 0 otherwise
boolean		1 if matrix is isotropic, 0 otherwise

Description The transformation class is defined by the transformation matrix, which operates as a premultiplier on column vectors containing homogeneous coordinates thus

$$\begin{bmatrix} x' & y' & z' & s' \end{bmatrix} = M \cdot \begin{bmatrix} x & y & z & s \end{bmatrix}'$$

where the conventional 3D coordinates are

$$\begin{bmatrix} \frac{x}{s} & \frac{y}{s} & \frac{z}{s} \end{bmatrix}$$

The matrix thus consists of

$$\begin{bmatrix} & & & T_x \\ & \mathbf{R} & & T_y \\ & & & T_z \\ 0 & 0 & 0 & S \end{bmatrix}$$

where $\mathbf{R}$ is a nonsingular transformation matrix, containing the rotation, reflection, nonuniform scaling, and shearing components; $\mathbf{T}$ is a translation vector; and S is a global scaling factor greater than zero.

VectorDouble

Supported Versions

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer	n	Number of elements
double[n]	d	Elements

Description This is a wrapper for a vector of doubles that can be used to store fields in the `Attribute` class.

See also `Attribute`

Supported Versions

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer	n	Number of elements
integer[n]	d	Elements

Description This is a wrapper for a vector of integers that can be used to store fields in the `Attribute` class.

See also `Attribute`

Supported Versions

Subtype of `Serializable`

Fields The class is defined by the following fields:

ENTITY/ OBJECT	VARIABLE	DESCRIPTION
integer	n	Number of elements
string[n]	d	Elements

Description This is a wrapper for a vector of strings that can be used to store fields in the `Attribute` class.

See also `Attribute`

Examples

To illustrate the use of the serialization format, some text files are listed in this session.

- A Mesh with Mixed Element Types
- A Planar Face

A Mesh with Mixed Element Types

A file containing a 3D mesh with 2nd order tetrahedral, prism and block elements. Some rows in the file are removed and replaced by an ellipsis(`...`).

```
# Created by COMSOL Multiphysics Fri Aug 26 12:43:42 2005


# Major & minor version
0 1
1 # number of tags
# Tags
7 fem35.0
1 # number of types
# Types
3 obj

# A mesh object

0 0 1
4 Mesh # class
3 # sdim
1503 # number of mesh points
0 # lowest mesh point index

# Mesh point coordinates
0 0 0
2.5 0 0
5 0 0

...
12.5 28.333330000000004 15
12.5 30 13.125
12.5 28.333330000000004 13.125

7 # number of element types

# Type #0
4 tet2 # type name

10 # number of nodes per element
318 # number of elements
```

```
# Elements
926 18 13 17 971 958 61 967 66 60
11 345 918 342 950 951 1137 949 373 1129
924 164 345 5 1026 1138 384 938 385 378
20 339 15 16 352 68 356 69 960 64
...
287 919 930 285 1100 1102 1152 317 1096 1098
3 227 4 8 936 28 243 35 945 36

30 # number of parameter values per element
0 # number of parameters
# Parameters
318 # number of domains
# Domains
1
...
1
1

0 # number of up/down pairs
# Up/down

# Type #1
6 prism2 # type name

18 # number of nodes per element
96 # number of elements
# Elements
85 174 90 476 474 557 221 118 237 499 1171 494 1170 1172 566 496 563 597
174 225 90 474 588 557 238 237 244 494 1173 579 1172 1174 566 598 597 604
174 175 225 474 472 588 222 238 239 494 1175 489 1173 1176 579 491 598 599
...
654 528 530 404 409 408 693 692 541 703 1333 538 1332 1334 543 460 459 465
654 504 528 404 405 409 687 693 694 703 1324 520 1333 1335 538 453 460 461
504 506 528 405 410 409 523 694 536 520 1336 525 1335 1337 538 462 461 466

54 # number of parameter values per element
0 # number of parameters
# Parameters

96 # number of domains
# Domains
2
2
2
...
2
2

0 # number of up/down pairs
# Up/down

# Type #2
4 hex2 # type name
```

```
27 # number of nodes per element
36 # number of elements
# Elements
410 506 409 528 762 760 831 859 525 466 1337 536 538 780 1339 775 1338 1343
1340 840 1341 853 777 837 1342 865 867
506 507 528 529 760 758 859 860 526 536 1281 537 539 775 1344 770 1340 1348
1345 853 1346 854 772 865 1347 866 868
...
893 785 809 787 718 719 722 723 908 910 1446 804 815 916 1487 801 1495 1502
1499 817 1500 806 747 750 1501 751 754

81 # number of parameter values per element
0 # number of parameters
# Parameters

36 # number of domains
# Domains
3
3
...
3
3

0 # number of up/down pairs
# Up/down

# Type #3
3 vtx # type name

1 # number of nodes per element
16 # number of elements
# Elements
0
4
20
...
723

0 # number of parameter values per element
0 # number of parameters
# Parameters

16 # number of domains
# Domains
0
1
...
14
15

0 # number of up/down pairs
# Up/down

# Type #4
```

```
4 edg2 # type name


3 # number of nodes per element
102 # number of elements
# Elements
4 9 37
9 14 50
14 19 63
...
175 174 222
174 85 221

3 # number of parameter values per element
102 # number of parameters
# Parameters
0 5 2.5
5 10 7.5
10 15 12.5
...
7.5 11.25 9.375
11.25 15 13.125

102 # number of domains
# Domains
0
0
...
27
27
27

0 # number of up/down pairs
# Up/down

# Type #5

4 tri2 # type name


6 # number of nodes per element
224 # number of elements
# Elements
18 17 13 66 61 60
164 5 345 385 384 378
20 339 15 352 68 356
...
404 409 405 460 453 461
405 409 410 461 462 466

12 # number of parameter values per element
0 # number of parameters
# Parameters

224 # number of domains
```

```
# Domains
12
7
7
...
5
5
5

224 # number of up/down pairs
# Up/down
1 0
1 0
1 0
1 0
...
2 0
2 0

# Type #6

5 quad2 # type name


9 # number of nodes per element
102 # number of elements
# Elements
85 90 476 557 118 499 1170 566 563
85 476 174 474 499 221 1171 496 494
...
809 787 722 723 815 817 1500 806 754
718 722 719 723 750 747 1501 754 751

18 # number of parameter values per element
0 # number of parameters
# Parameters

102 # number of domains
# Domains
1
4
4
...
8
2

102 # number of up/down pairs
# Up/down
2 0
2 0
...
3 0
3 0
```

A Planar Face

The following listing is a complete representation of a planar 3D face.

```
# Created by COMSOL Multiphysics Fri Aug 26 14:19:54 2005

# Major & minor version
0 1
######### Tags
1 # number of tags
# Tags
2 b1

######## Types
1 # number of types
# Types
3 obj

######## Records

# A planar face object

0 0 1
5 Geom3 # class
1 # version
2 # type
1e-010 # gtol
0.0001 # resTol
4 # number of vertices
# Vertices
# X Y Z sub tol
0 0 0 -1 1e-010
1 0 6.1257422745431001e-017 -1 1e-010
0 1 0 -1 1e-010
1 1 6.1257422745431001e-017 -1 1e-010
4 # number of parameter vertices
# Parameter vertices
# vtx s t fac mfd tol
1 0 0 -1 1 NAN
2 1 0 -1 1 NAN
3 0 1 -1 1 NAN
4 1 1 -1 1 NAN

4 # number of edges
# Edges
# vtx1 vtx2 s1 s2 sub mfd tol
1 2 0 1 -1 2 NAN
2 4 0 1 -1 3 NAN
4 3 0 1 -1 4 NAN
3 1 0 1 -1 5 NAN
4 # number of parameter edges
# Parameter edges
# edg v1 v2 s1 s2 up down mfdfac mfd tol
1 1 2 0 1 1 0 1 1 NAN
2 2 4 0 1 1 0 2 1 NAN
```

```
3 4 3 0 1 1 0 3 1 NAN
4 3 1 0 1 1 0 4 1 NAN

1 # number of faces
# Faces
# up down mfd tol
0 0 1 NAN

5 # number of 3D manifolds
# Manifold #0
5 Plane # class
0 0 # version
3 # sdim
0 3 1 # transformation
0 0 0 # root point
-6.1257422745431001e-017 0 1 # normal
1 0 6.1257422745431001e-017 # derivatives
0 # degenerated?

# Manifold #1
8 Straight # class
0 0 # version
3 # sdim
0 3 1 # transformation
0 0 0 # root point
1 0 6.1257422745431001e-017 # direction
1 # parameter scale

# Manifold #2
8 Straight # class
0 0 # version
3 # sdim
0 3 1 # transformation
1 0 6.1257422745431001e-017 # root point
0 1 0 # direction
1 # parameter scale

# Manifold #3
8 Straight # class
0 0 # version
3 # sdim
0 3 1 # transformation
1 1 6.1257422745431001e-017 # root point
-1 0 -6.1257422745431001e-017 # direction
1 # parameter scale

# Manifold #4
8 Straight # class
0 0 # version
3 # sdim
0 3 1 # transformation
0 1 0 # root point
0 -1 0 # direction
1 # parameter scale
```

```
4 # number of parameter curves
# Paramerer curve #0
8 PolChain # class
0 0 # version
2 # sdim
0 2 1 # transformation
2 2 0 0 1 0 # chain


# Paramerer curve #1
8 PolChain # class
0 0 # version
2 # sdim
0 2 1 # transformation
2 2 1 0 1 1 # chain


# Paramerer curve #2
8 PolChain # class
0 0 # version
2 # sdim
0 2 1 # transformation
2 2 1 1 0 1 # chain


# Paramerer curve #3
8 PolChain # class
0 0 # version
2 # sdim
0 2 1 # transformation
2 2 0 1 0 0 # chain

# Attributes
0 # nof attributes
```

Index