

COMSOL MULTIPHYSICS®

USER'S GUIDE

VERSION 3.5



How to contact COMSOL:**Benelux**

COMSOL BV
Röntgenlaan 19
2719 DX Zoetermeer
The Netherlands
Phone: +31 (0) 79 363 4230
Fax: +31 (0) 79 361 4212
info@comsol.nl
www.comsol.nl

Denmark

COMSOL A/S
Diplomvej 376
2800 Kgs. Lyngby
Phone: +45 88 70 82 00
Fax: +45 88 70 80 90
info@comsol.dk
www.comsol.dk

Finland

COMSOL OY
Arabianranta 6
FIN-00560 Helsinki
Phone: +358 9 2510 400
Fax: +358 9 2510 4010
info@comsol.fi
www.comsol.fi

France

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

Germany

COMSOL Multiphysics GmbH
Berliner Str. 4
D-37073 Göttingen
Phone: +49-551-99721-0
Fax: +49-551-99721-29
info@comsol.de
www.comsol.de

Italy

COMSOL S.r.l.
Via Vittorio Emanuele II, 22
25122 Brescia
Phone: +39-030-3793800
Fax: +39-030-3793899
info.it@comsol.com
www.it.comsol.com

Norway

COMSOL AS
Søndre gate 7
NO-7485 Trondheim
Phone: +47 73 84 24 00
Fax: +47 73 84 24 01
info@comsol.no
www.comsol.no

Sweden

COMSOL AB
Tegnérsgatan 23
SE-111 40 Stockholm
Phone: +46 8 412 95 00
Fax: +46 8 412 95 10
info@comsol.se
www.comsol.se

Switzerland

FEMLAB GmbH
Technoparkstrasse 1
CH-8005 Zürich
Phone: +41 (0)44 445 2140
Fax: +41 (0)44 445 2141
info@femlab.ch
www.femlab.ch

United Kingdom

COMSOL Ltd.
UH Innovation Centre
College Lane
Hatfield
Hertfordshire AL10 9AB
Phone: +44-(0)-1707 636020
Fax: +44-(0)-1707 284746
info.uk@comsol.com
www.uk.comsol.com

United States

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

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

COMSOL, Inc.
744 Cowper Street
Palo Alto, CA 94301
Phone: +1-650-324-9935
Fax: +1-650-324-9936

info@comsol.com
www.comsol.com

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

Company home page
www.comsol.com

COMSOL user forums
www.comsol.com/support/forums

COMSOL Multiphysics User's Guide

© COPYRIGHT 1994–2008 by COMSOL AB. All rights reserved

Patent pending

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from COMSOL AB.

COMSOL, COMSOL Multiphysics, COMSOL Script, COMSOL Reaction Engineering Lab, and FEMLAB are registered trademarks of COMSOL AB.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Version: September 2008 COMSOL 3.5

Part number: CM020002

C O N T E N T S

Chapter 1: Introduction

The Documentation Set	2
Typographical Conventions	4
About COMSOL Multiphysics	5
The COMSOL Multiphysics Environment	8
The COMSOL Modules	12
The AC/DC Module	12
The Acoustics Module	13
The Chemical Engineering Module	14
The Earth Science Module	15
The Heat Transfer Module	16
The MEMS Module	17
The RF Module	17
The Structural Mechanics Module	18
The CAD Import Modules	20
Internet Resources	21
COMSOL Web Sites	21
COMSOL User Forums	21

Chapter 2: Geometry Modeling and CAD Tools

The COMSOL Multiphysics Geometry and CAD Environment	24
Overview of Geometry Modeling Concepts	24
Setting up Axes and Grid	25
Creating Cartesian and Cylindrical Coordinate Systems	27
The Status Bar	30

Creating Composite Geometry Objects	31
Moving, Rotating, Scaling, and Mirroring Geometry Objects.	33
Creating an Array of Geometry Objects	34
Copying and Pasting Geometry Objects	34
Coercing Geometry Objects	34
Removing Interior Boundaries	35
Splitting Geometry Objects	35
Geometry Domain Names	35
Viewing Geometries	36
Entering Draw Mode	36
Creating a 1D Geometry Model	37
Creating a 2D Geometry Model	39
Using the 2D Draw Toolbar and the Draw Menu.	39
2D Solid Modeling Techniques	42
2D Boundary Modeling	46
Creating Fillets and Chamfers	49
Drawing Tangents	52
Editing Geometry Objects	54
Creating a 3D Geometry Model	56
Using the 3D Draw Toolbar and Draw Menu	56
Creating 3D Geometries from 2D Geometric Objects	59
Creating and Using 2D Work Planes	59
Extruding, Revolving, and Embedding	63
Solid Modeling Using Solid Objects and Boolean Operations	67
Editing 3D Objects	75
Exploring Geometric Properties	77
Introduction	77
Using The Geometric Properties Dialog Box	77
Displaying Geometric Properties in the Message Log	81
Centroid, Center of Gravity, and Distance in Electronic Conductor.	81
Importing and Exporting Geometry Objects and CAD Models	85
Importing and Exporting COMSOL Multiphysics Files	86
Importing and Exporting DXF	87

Importing STL and VRML Files	89
Creating a Geometry for Successful Analysis	92
Using Assemblies	92
Using Symmetries	93
Removing Unnecessary Boundaries	93
Making the Geometry Match the Boundary Conditions	93
Avoiding Excessively Small Details, Holes, and Gaps.	93
Avoiding Singularities and Degeneracies in the Geometry	94
Associative Geometry	94

Chapter 3: Visualization and Selection Tools

Visualizing a Model	98
The Visualization/Selection Toolbar	99
Zooming	100
Changing the View in 3D.	101
Using Lighting	104
Changing the View to a Different Axis Plane	112
Using Transparency in 3D Plots	112
Controlling the Resolution of the Visualization Mesh	113
Adding Labels to Domains	114
Rendering and Highlighting Geometry Objects	116
Scaling of Load and Constraint Symbols	119
Saving Preferences for Labels, Rendering, and Highlighting	119
Suppressing the Display of Objects	122
Rendering Large Geometry Objects	123
Viewing Multiple Geometries	124
Selecting Domains	126
General Object Selection Methods	126
Object Selection Methods in 2D	127
Object Selection Methods in 3D	127

Chapter 4: Modeling Physics and Equations

Variables and Expressions	138
Using Variables and Expressions	138
Variable Classification and Geometric Scope	138
Defining Constants, Expressions, and Coupling Variables	139
Specifying Varying Coefficients and Material Properties	143
Copying Physics Settings	144
Entering Vector-Valued Expressions	145
Using Mathematical and Logical Operators and Functions	147
Specifying Discontinuous Functions	149
Using Analytic Functions	150
Piecewise Analytic Functions	151
Using Functions Based on Interpolated Data	154
Using COMSOL Script or MATLAB Functions	161
Function Input/Output Considerations	161
Using Special Operators	163
Variable Naming Conventions	169
Geometric Variables	170
Using Field Variables	176
Field Variable Definition and Categories	176
Shape Function Variables	177
Application Mode Variables	180
Expression Variables	181
Coupling Variables	181
Special Variables	181
Specifying Model Settings	185
Generating Variables	185
Simplifying Expressions	186
Selecting an Equation System Form	186
Selecting a Geometry Shape Order	187
Model Settings with Multiple Frames	187
Using Units	188
Unit Systems in COMSOL Multiphysics	188

Specifying the Unit System	189
Using the Unit Syntax	192
Switching Unit System	194
About Temperature Units	195
SI Base Units, Derived Units, and Additional Units	196
Special British Engineering Units	199
Special CGSA Units	200
Special EMU Units	201
Special ESU Units.	202
Special FPS Units	203
Special IPS Units	203
Special MPa Units.	203
Special Gravitational IPS Units	204
Specifying Physics Settings	205
Opening the Physics Settings Dialog Boxes	205
Working with Domain Groups.	206
Specifying Subdomain Settings and PDE Coefficients	210
Using the Subdomain Settings Dialog Box	210
Specifying PDE Coefficients and Material Properties	212
Specifying Initial Conditions	214
Specifying the Finite Element Type	216
Specifying Weak Terms	217
Ideal and Non-Ideal Constraints	217
Viewing and Modifying the Full Equation System	218
Modeling Anisotropic Materials	221
Using the Model Tree	223
The Model Tree Views.	223
Using the Overview View	224
Using the Detail View	225
Using the Inspect View	227
Model Tree Settings.	228
Using the Materials/Coefficients Library	229
Using Material/Coefficient Data in Models	229
Accessing Material Properties	232

Using Your Own Material Data	233
Using MatWeb Material Property Data	235
Interpolation of External Data and Nonlinear Materials	236
Modeling Thermal Conductivity from Measurements	236
Example: Importing Unstructured 3D Data from an External Source	240
Specifying Boundary Conditions	246
Interior and Exterior Boundaries	246
Setting Conditions on Boundaries	247
Modifying Boundary Settings for the Equation System	249
Setting Conditions on Interior Boundaries	252
Specifying Boundary Conditions for Identity Pairs	253
Specifying Point and Edge Settings	254
Using Point Settings	254
Using Edge Settings	255
Setting Point and Edge Constraints	256
Using Periodic Boundary Conditions	257
Periodic Boundary Condition Example.	261
Specifying Application Scalar Variables	265
Computing Accurate Fluxes	266
Flux Computation Example	267
Using Coupling Variables	269
Integration Coupling Variables	269
Integration Coupling Variables—Examples	272
Extrusion Coupling Variables	275
Extrusion Coupling Variables—Example	280
Projection Coupling Variables	284
Projection Coupling Variables—Example	290
General Issues When Using Coupling Variables	294
Using Boundary Distance Variables	295
Plotting Distances to Walls in Backstep Model.	296

Chapter 5: Meshing

Creating Meshes	300
Mesh Elements	300
Meshing Techniques	300
Creating Free Meshes	301
Meshing Progress Indication	318
Creating Mapped Meshes in 2D	320
Extruding and Revolving 2D Meshes	328
Creating Swept Meshes in 3D	333
Creating Boundary Layer Meshes	338
Meshing Thin Structures	342
Mesh Statistics	343
Refining Meshes	344
Refinement Methods	344
Refining Selected Elements	344
Converting Mesh Elements	345
Converting Elements	345
Interactive Meshing	346
Using the Mesh Toolbar	346
Deleting the Mesh	349
Meshing a Domain Without Remeshing Boundaries	349
Undoing Meshing Operations	351
Creating a Combined Free and Mapped Mesh in 2D	351
Creating Swept Meshes Interactively	353
Creating Boundary Layer Meshes Interactively	359
Copying Meshes	360
Importing and Exporting Meshes	364
Using COMSOL Multiphysics Files	364
Importing NASTRAN Files	364

Creating a Geometry from a Mesh	369
Visualizing Meshes	370
Visualizing the Mesh Quality	371
Mesh Cases	373
Avoiding Inverted Mesh Elements	374
Inverted Mesh Elements	374
Using Linear Geometry Shape Order	374
Modifying the Geometry or Mesh	375
Visualizing Inverted Mesh Elements	375

Chapter 6: Solving the Model

Selecting a Solver	378
The COMSOL Multiphysics Solvers	378
Selecting Analysis Types	378
Selecting a Stationary, a Time-Dependent, or an Eigenvalue Solver	380
Linear Solvers vs. Nonlinear Solvers	380
Remarks on Solver-Related Model Characteristics	382
Setting Up Parametric Sweeps	383
Setting Up Geometric Parametric Sweeps	384
The Solvers and Their Settings	385
The Stationary Solver	385
The Time-Dependent Solver	391
The Eigenvalue Solver	402
The Parametric Solver	405
The Stationary Segregated Solver	411
The Parametric Segregated Solver	416
The Time-Dependent Segregated Solver	417
Adaptive Mesh Refinement	419
Sensitivity Analysis and Optimization	423
Plotting the Solution While Solving	424
Reference	425

The Linear System Solvers	426
Selecting a Linear System Solver	426
Linear System Solver Settings	428
Linear System Solver Selection Guidelines	428
Selecting a Preconditioner	429
Preconditioner Selection Guidelines	430
Guidelines for Classifying Your PDE Problem	431
The Preconditioned Linear System	435
The Solver Manager and Solver Scripting	436
Initial Value Settings	436
Choosing Variables for Which to Solve	439
Selecting Variables for Solution Output	441
Solver Scripting	443
Computing the Solution	445
Starting and Restarting the Solvers	445
Updating a Model.	445
Getting the Initial Value	446
Solution Progress.	446
Monitoring Memory Usage	452

Chapter 7: Postprocessing and Visualization

Postprocessing Results	454
COMSOL Multiphysics Postprocessing Overview	454
Postprocessing Expressions	454
Selection of Frames for Postprocessing	455
Interpolating Expressions at Arbitrary Points	456
Computing Quantities Using Integration	456
Getting Numerical Results Directly	459
Plotting Global Expressions	461
Visualization Using the Plot Parameters Dialog Box.	462
Cross-Section Plots	473
Domain Plots	475
Specifying Line and Surface Properties	476

Creating Animations	478
Plotting Values While Solving	481
General Postprocessing Settings	487
Exporting Postprocessing Data to a File	487
Exporting the Current Plot to a File.	492
Exporting Data to COMSOL Script or MATLAB	492
Exporting Images	493
Using Max/Min Markers	496
Postprocessing 2D Axisymmetric Models in 3D	496
Postprocessing Results in 1D	497
1D Postprocessing Overview	497
1D Standard Plots	497
1D Cross-Section Plots	498
1D Domain Plots	499
Postprocessing Results in 2D	501
2D Postprocessing Overview	501
Using the 2D Plot Toolbar	501
2D Standard Plots	502
2D Cross-Section Plots	526
2D Domain Plots	528
Postprocessing Results in 3D	532
3D Postprocessing Overview	532
Using the 3D Plot Toolbar	532
3D Standard Plots	534
3D Cross-Section Plots	544
3D Domain Plots	548
The Figure Window	551
The Figure Window Toolbar	551
The Edit Plot Dialog Box	552
Using the Report Generator	555
Report Generator Overview	555
Creating a Report	556
Report Contents	559

Chapter 8: Model Maintenance and Model Libraries

Opening and Saving COMSOL Multiphysics Models	568
Model Files Formats.	568
Opening a COMSOL Multiphysics Model.	568
Saving a COMSOL Multiphysics Model.	569
Saving on Server Computer.	570
Model MPH-files	570
Setting Model Properties.	571
Model M-Files	574
Using Model M-files.	574
Using the COMSOL Multiphysics Model Library	576
Models in the Model Library	576
Creating a User Model Library	577
Saving Models for a User Model Library	577

Chapter 9: Glossary

Glossary of Terms	580
INDEX	603

Introduction

Welcome to COMSOL Multiphysics®! This *User's Guide* details features and techniques that help you throughout all of your COMSOL Multiphysics modeling. In this book, for example, we detail procedures to build model geometries in COMSOL Multiphysics, create a mesh for the finite elements, create variables and expressions you can use within a model, and solve and postprocess the solution. The explanations, tutorials, and examples show you, step by step, how to tap into many functions and capabilities available in the COMSOL environment.

This introductory chapter provides an overview of COMSOL Multiphysics and its product family.

The Documentation Set

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

- *COMSOL Quick Installation Guide*—basic information for installing the COMSOL software and getting started. Included in the DVD package.
- *COMSOL New Release Highlights*—information about new features and models in the 3.5 release. Included in the DVD package.
- *COMSOL License Agreement*—the license agreement. Included in the DVD package.
- *COMSOL Installation and Operations Guide*—besides covering various installation options, it describes system requirements and how to configure and run the COMSOL software on different platforms.
- *COMSOL Multiphysics Quick Start and Quick Reference*—provides a quick overview of COMSOL Multiphysics' capabilities and how to access them. A reference section contains comprehensive lists of predefined variable names, mathematical functions, COMSOL Multiphysics operators, equation forms, and application modes.
- *COMSOL Multiphysics User's Guide*—the book you are reading, it covers the functionality of COMSOL Multiphysics across its entire range from geometry modeling to postprocessing. It serves as a tutorial and a reference guide to using COMSOL Multiphysics.
- *COMSOL Multiphysics Modeling Guide*—provides an in-depth examination of the software's application interfaces and how to use them to model different types of physics and to perform equation-based modeling using PDEs.
- *COMSOL Multiphysics Model Library*—consists of a collection of ready-to-run models that cover many classic problems and equations from science and engineering. These models have two goals: to show the versatility of COMSOL Multiphysics and the wide range of applications it covers; and to form an educational basis from which you can learn about COMSOL Multiphysics and also gain an understanding of the underlying physics.

- *COMSOL Multiphysics Scripting Guide*—shows how to access all of COMSOL Multiphysics’ capabilities from the COMSOL Script environment and the MATLAB programming environment.
- *COMSOL Multiphysics Reference Guide*—provided only in HTML and PDF versions. This book reviews each command that lets you access COMSOL Multiphysics functions from within COMSOL Script and MATLAB. Additionally, it describes some advanced features and settings in COMSOL Multiphysics and provides background material and references.

In addition, each of the optional modules

- *AC/DC Module*
- *Acoustics Module*
- *Chemical Engineering Module*
- *Earth Science Module*
- *Heat Transfer Module*
- *MEMS Module*
- *RF Module*
- *Structural Mechanics Module*

comes with its own *User’s Guide* and *Model Library*. Many modules also include a *Reference Guide*, available in HTML and PDF versions.

The documentation for the optional CAD Import Module is available in the *CAD Import Module User’s Guide*, and the documentation for the optional Material Library in the *Material Library User’s Guide*.

COMSOL Script comes with a *COMSOL Script User’s Guide* as well as a comprehensive *COMSOL Script Command Reference* and its add-on labs—the COMSOL Reaction Engineering Lab, Optimization Lab, and Signals & Systems Lab—include documentation in separate *User’s Guides*. The COMSOL Reaction Engineering Lab also includes a comprehensive *Model Library*.

Note: The full documentation set is available in electronic versions—as PDF files and Help Desk format—after installation.

Typographical Conventions

All COMSOL manuals use a set of consistent typographical conventions that should make it easy for you to follow the discussion, realize what you can expect to see on the screen, and know which data you must enter into various data-entry fields. In particular, you should be aware of these conventions:

- A **boldface** font of the shown size and style indicates that the given word(s) appear exactly that way on the COMSOL graphical user interface (for toolbar buttons in the corresponding tooltip). For instance, we often refer to the **Model Navigator**, which is the window that appears when you start a new modeling session in COMSOL; the corresponding window on the screen has the title **Model Navigator**. As another example, the instructions might say to click the **Multiphysics** button, and the boldface font indicates that you can expect to see a button with that exact label on the COMSOL user interface.
- The names of other items on the graphical user interface that do not have direct labels contain a leading uppercase letter. For instance, we often refer to the Draw toolbar; this vertical bar containing many icons appears on the left side of the user interface during geometry modeling. However, nowhere on the screen will you see the term “Draw” referring to this toolbar (if it were on the screen, we would print it in this manual as the **Draw** menu).
- The symbol **>** indicates a menu item or an item in a folder in the **Model Navigator**. For example, **Physics>Equation System>Subdomain Settings** is equivalent to: On the **Physics** menu, point to **Equation System** and then click **Subdomain Settings**. **COMSOL Multiphysics>Heat Transfer>Conduction** means: Open the **COMSOL Multiphysics** folder, open the **Heat Transfer** folder, and select **Conduction**.
- A Code (monospace) font indicates keyboard entries in the user interface. You might see an instruction such as “Type 1.25 in the **Current density** edit field.” The monospace font also indicates COMSOL Script codes.
- 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 books in the COMSOL documentation set also appear using an italic font.

About COMSOL Multiphysics

COMSOL Multiphysics is a powerful interactive environment for modeling and solving all kinds of scientific and engineering problems based on partial differential equations (PDEs). With this software you can easily extend conventional models for one type of physics into multiphysics models that solve coupled physics phenomena—and do so simultaneously. Accessing this power does not require an in-depth knowledge of mathematics or numerical analysis. Thanks to the built-in *physics modes* it is possible to build models by defining the relevant physical quantities—such as material properties, loads, constraints, sources, and fluxes—rather than by defining the underlying equations. You can always apply these variables, expressions, or numbers directly to solid domains, boundaries, edges, and points independently of the computational mesh. COMSOL Multiphysics then internally compiles a set of PDEs representing the entire model. You access the power of COMSOL Multiphysics as a standalone product through a flexible graphical user interface, or by script programming in the COMSOL Script language or in the MATLAB language.

Using these application modes, you can perform various types of analysis including:

- Stationary and time-dependent analysis
- Linear and nonlinear analysis
- Eigenfrequency and modal analysis

When solving the models, COMSOL Multiphysics uses the proven *finite element method (FEM)*. The software runs the finite element analysis together with adaptive meshing and error control using a variety of numerical solvers. A more detailed description of this mathematical and numerical foundation appears in the *COMSOL Multiphysics User's Guide* and in the *COMSOL Multiphysics Modeling Guide*.

PDEs form the basis for the laws of science and provide the foundation for modeling a wide range of scientific and engineering phenomena. Therefore you can use COMSOL Multiphysics in many application areas, just a few examples being:

- Acoustics
- Bioscience
- Chemical reactions
- Diffusion
- Electromagnetics

- Fluid dynamics
- Fuel cells and electrochemistry
- Geophysics
- Heat transfer
- Microelectromechanical systems (MEMS)
- Microwave engineering
- Optics
- Photonics
- Porous media flow
- Quantum mechanics
- Radio-frequency components
- Semiconductor devices
- Structural mechanics
- Transport phenomena
- Wave propagation

Many real-world applications involve simultaneous couplings in a system of PDEs—*multiphysics*. For instance, the electric resistance of a conductor often varies with temperature, and a model of a conductor carrying current should include resistive-heating effects. The *COMSOL Multiphysics Quick Start and Quick Reference* provides an introduction to multiphysics modeling in the section “A Quick Tour of COMSOL Multiphysics” on page 19. In addition, the *COMSOL Multiphysics Modeling Guide* covers multiphysics modeling techniques in the section “Creating Multiphysics Models”. The “Multiphysics” chapter in the *COMSOL Multiphysics Model Library* also contains several examples. Many predefined multiphysics couplings provide easy-to-use entry points for common multiphysics applications.

In its base configuration, COMSOL Multiphysics offers modeling and analysis power for many application areas. For several of the key application areas we also provide optional modules. These application-specific modules use terminology and solution methods specific to the particular discipline, which simplifies creating and analyzing models. The COMSOL 3.5 product family includes the following modules:

- AC/DC Module
- Acoustics Module
- Chemical Engineering Module

- Earth Science Module
- Heat Transfer Module
- MEMS Module
- RF Module
- Structural Mechanics Module

The CAD Import Module provides the possibility to import CAD data using the following formats: IGES, SAT (Acis), Parasolid, and Step. Additional add-ons provide support for CATIA V4, CATIA V5, Pro/ENGINEER, Autodesk Inventor, and VDA-FS. It also supports bidirectional interfaces with SolidWorks and Autodesk Inventor.

You can build models of all types in the COMSOL Multiphysics user interface. For additional flexibility, COMSOL also provides its own scripting language, COMSOL Script, where you can access the model as a Model M-file or a data structure. COMSOL Multiphysics also provides a seamless interface to MATLAB. This gives you the freedom to combine PDE-based modeling, simulation, and analysis with other modeling techniques. For instance, it is possible to create a model in COMSOL and then export it to Simulink as part of a control-system design.

We are delighted you have chosen COMSOL Multiphysics for your modeling needs and hope that it exceeds all expectations. Thanks for choosing COMSOL!

The COMSOL Multiphysics Environment

This section describes the major components in the COMSOL Multiphysics environment.

When starting COMSOL Multiphysics, you are greeted by the *Model Navigator*. Here you begin the modeling process and control all program settings. It lets you select space dimension and application modes to begin working on a new model, open an existing model you have already created, or open an entry in the Model Library.

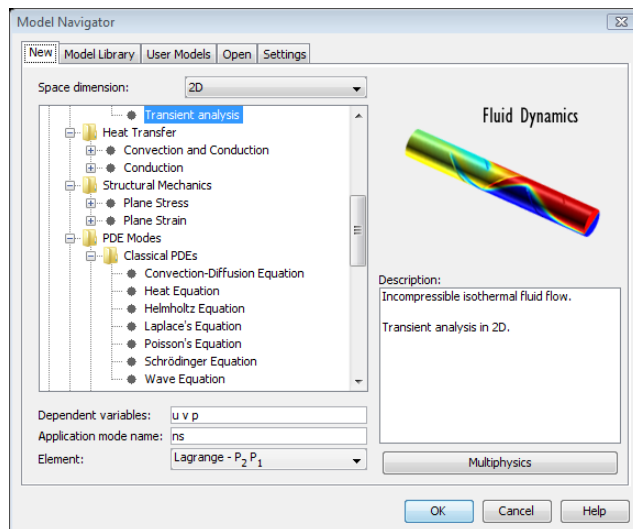


Figure 1-1: The COMSOL Multiphysics Model Navigator.

COMSOL Multiphysics provides an integrated graphical user interface where you can build and solve models by using predefined *physics modes*, *PDE modes*, or a combination of them—multiphysics modeling.

These application modes are analogous to templates in that you define material properties, boundary conditions, and other quantities; COMSOL Multiphysics then

creates the PDEs. Application modes supply models for performing studies in areas such as:

- Acoustics
- Diffusion
- Electromagnetics
- Fluid mechanics
- Heat transfer
- Structural mechanics
- Optimization¹ and sensitivity analysis
- PDEs

2D, 3D, and axisymmetric variants are available for most application modes. You can find extensive information about the application modes in the *COMSOL Multiphysics Modeling Guide*.

To illustrate the uses of these application modes and other ways to put COMSOL Multiphysics to work, we include prewritten ready-to-run models of familiar and interesting problems in the Model Library. This consists of two elements: model files installed with COMSOL Multiphysics and a dedicated manual (the *COMSOL Multiphysics Model Library*). Simply load one of them into the Model Navigator to view the models settings and the solution. As noted earlier, each model includes extensive documentation complete with technical background, a discussion of the results, and step-by-step descriptions of how to set up, solve, and postprocess the model.

In a short time, though, you will want to start creating your own models. An important part of the process is creating the geometry. The COMSOL Multiphysics user interface contains a set of CAD tools for geometry modeling in 1D, 2D, and 3D. The package can also import geometries using the DXF, STL, VRML, and COMSOL file formats. COMSOL Multiphysics can also directly import 3D meshes on NASTRAN format. In combination with the programming tools, you can even use images and magnetic resonance imaging (MRI) data to create a geometry.

In creating models you also have great flexibility in setting up various constants and variables using a number of variables as well as mathematical and logical functions. In the COMSOL programming environment you can work with the COMSOL Script

1. Optimization in COMSOL Multiphysics requires the Optimization Lab.

language or the MATLAB language to define material properties, loads, sources, and boundary conditions.

When the geometry is complete and the parameters are defined, COMSOL Multiphysics automatically meshes the geometry. However, you can take charge of the mesh-generation process through a set of control parameters.

Next comes the solution stage. Here COMSOL Multiphysics comes with a suite of solvers, all developed by leading experts, for stationary, eigenvalue, and time-dependent problems. For solving linear systems, the software features both direct and iterative solvers. A range of preconditioners are available for the iterative solvers. COMSOL Multiphysics sets up solver defaults appropriate for the chosen application mode and automatically detects linearity and symmetry in the model. A segregated solver provides efficient solution schemes for large multiphysics models, turbulence modeling, and other challenging applications. You can record sequential solver calls to a solver script that you can further edit and run to automate the solution process.

For postprocessing, COMSOL Multiphysics provides tools for plotting and postprocessing any model quantity or parameter:

- Surface plots
- Slice plots
- Isosurfaces
- Contour plots
- Deformed shape plots
- Arrow plots
- Streamline plots and particle tracing
- Cross-sectional plots
- Animations
- Data display and interpolation
- Integration on boundaries and subdomains

In addition, you can continue the postprocessing in the COMSOL Script environment, where you can work with COMSOL Multiphysics data structures and functions using script programming.

To document your models, the COMSOL Multiphysics Report Generator provides a comprehensive report of the entire model, including graphics of the geometry, mesh,

and postprocessing quantities. You can print the report directly or save it as an HTML file for viewing through a web browser and further editing.

The COMSOL Modules

The optional modules

- AC/DC Module
- Acoustics Module
- Chemical Engineering Module
- Earth Science Module
- Heat Transfer Module
- MEMS Module
- RF Module
- Structural Mechanics Module

are optimized for specific application areas. They offer discipline-standard terminology and interfaces, materials libraries, specialized solvers, elements, and visualization tools.

The AC/DC Module

The AC/DC Module provides a unique environment for simulation of AC/DC electromagnetics in 2D and 3D. The AC/DC Module is a powerful tool for detailed analysis of coils, capacitors, and electrical machinery. With this module you can run static, quasi-static, transient, and time-harmonic simulations in an easy-to-use graphical user interface.

The available application modes cover the following types of electromagnetics field simulations:

- Electrostatics
- Conductive media DC
- Magnetostatics
- Low-frequency electromagnetics

Material properties include inhomogeneous and fully anisotropic materials, media with gains or losses, and complex-valued material properties. Infinite elements makes it possible to model unbounded domains. In addition to the standard postprocessing features, the AC/DC Module supports direct computation of lumped parameters such as capacitances and inductances as well as electromagnetic forces and torques. With the

multiphysics capabilities of COMSOL Multiphysics, you can couple simulations with heat transfer, structural mechanics, fluid flow formulations, and any other physical phenomena.

This module also provides interfaces for integrating SPICE circuits and importing ECAD drawings.

The Acoustics Module

The Acoustics Module provides tailored interfaces for modeling of acoustics in fluids and solids. The module supports time-harmonic, modal, and transient analyses for fluid pressure as well as static, transient, eigenfrequency, and frequency-response analyses for structures. The available application modes include:

- Pressure acoustics
- Aeroacoustics (acoustics in an ideal gas with an irrotational mean flow)
- Compressible irrotational flow
- Plane strain, axisymmetric stress/strain, and 3D stress/strain

For the pressure acoustics applications, you can choose to analyze the scattered wave in addition to the total wave. PMLs (perfectly matched layers) provide accurate simulations of open pipes and other models with unbounded domains. The modeling domain can include dipole sources as well as monopole sources, and it is easy to specify point sources in terms of flow, intensity, or power. The module also includes modeling support for several types of damping. For postprocessing of pressure acoustics models, you can compute the far field.

Typical application areas for the Acoustics Module include:

- Automotive applications such as mufflers and car interiors
- Modeling of loudspeakers and microphones
- Aeroacoustics
- Underwater acoustics

Using the full multiphysics couplings within the COMSOL Multiphysics environment, you can couple the acoustic waves to, for example, an electromagnetic analysis or a structural analysis for acoustic-structure interaction.

The Chemical Engineering Module

The Chemical Engineering Module presents a powerful way of modeling equipment and processes in chemical engineering. It provides customized interfaces and formulations for momentum, mass, and heat transport coupled with chemical reactions for applications such as:

- Reaction engineering and design
- Heterogeneous catalysis
- Separation processes
- Fuel cells and industrial electrolysis
- Process control together with Simulink

COMSOL Multiphysics excels in solving systems of coupled nonlinear PDEs that can include:

- Heat transfer
- Mass transfer through diffusion, convection, and migration
- Fluid dynamics
- Chemical reaction kinetics
- Varying material properties

The multiphysics capabilities of COMSOL Multiphysics can fully couple and simultaneously model fluid flow, mass and heat transport, and chemical reactions.

In fluid dynamics you can model fluid flow through porous media, characterize flow with the incompressible Navier-Stokes equations (for turbulent or laminar flow), and simulate nonisothermal flow and multiphase flow. It is easy to represent chemical reactions by source or sink terms in mass and heat balances. These terms can be of arbitrary order. All formulations exist for both Cartesian and cylindrical coordinates (for axisymmetric models) as well as for stationary and time-dependent cases.

The available application modes are:

- Momentum balances
 - Incompressible Navier-Stokes equations
 - Darcy's law
 - Brinkman equations
 - Non-Newtonian flow
 - Nonisothermal and weakly compressible flow
 - Turbulent flow, k- ϵ turbulence model
 - Turbulent flow, k- ω turbulence model
 - Multiphase flow: Bubbly Flow, Mixture Model, and Two-Phase Flow using both the level set method and the phase field method.
- Energy balances
 - Heat conduction
 - Heat convection and conduction
- Mass balances
 - Diffusion
 - Convection and diffusion
 - Electrokinetic flow
 - Maxwell-Stefan diffusion and convection
 - Nernst-Planck transport equations

The Earth Science Module

The earth and planets are giant laboratories that involve all manner of physics. The Earth Science Module combines application modes for fundamental processes and links to COMSOL Multiphysics and the other modules for structural mechanics and electromagnetics analyses. New physics represented include heating from radiogenic decay that produces the geotherm, which is the increase in background temperature with depth. The variably saturated flow application modes analyze unsaturated zone processes (important to environmentalists) and two-phase flow (of particular interest in the petroleum industry as well as steam-liquid systems). Important in earth sciences, the heat transfer and chemical transport application modes explicitly account for physics in the liquid, solid, and gas phases.

Available application modes are:

- Darcy’s law for hydraulic head, pressure head, and pressure. Also part of a predefined interface for poroelasticity (requires the Structural Mechanics Module or the MEMS Module).
- Solute transport in saturated and variably saturated porous media
- Richards’ equation including nonlinear material properties using van Genuchten, Brooks and Carey, or user-defined parameters.
- Heat transfer by conduction and convection in porous media with one mobile fluid, one immobile fluid, and up to five solids
- Brinkman equations
- Incompressible Navier-Stokes equations

The Earth Science Module Model Library contains a number of interesting examples, both single physics and multiphysics. Poroelasticity illustrates two-way link solid displacement and movement of fluids in pore spaces. The volcano model combines fluid flow with electromagnetics application modes. Several examples demonstrate interpolation from experimental data. Multiple flow laws are linked in the two-phase flow and Darcy–Brinkman–Navier-Stokes example near a well. This module combines new and existing physics in a form that earth scientists can readily use.

The Heat Transfer Module

The Heat Transfer Module supports all fundamental mechanisms of heat transfer, including *conductive*, *convective*, and *radiative* heat transfer (both *surface-to-surface* and *surface-to-ambient* radiation). Using the application modes in this module along with inherent multiphysics capabilities of COMSOL Multiphysics, you can model a temperature field in parallel with other physics—a powerful combination that makes your models even more accurate and representative of the real world.

Available application modes are:

- General heat transfer, including conduction, convection, and surface-to-surface radiation
- Bioheat equation for heat transfer in biomedical systems
- Highly conductive layer for modeling of heat transfer in thin structures
- Nonisothermal flow application mode for nonisothermal incompressible fluid flow
- Turbulent flow using the k - ϵ and k - ω turbulence models

The Heat Transfer Module Model Library contains models, many with multiphysics couplings, that cover applications in electronics and power systems, process industries, and manufacturing industries. This Model Library also provides tutorial and benchmark models.

The MEMS Module

One of the most exciting areas of technology to emerge in recent years is MEMS (microelectromechanical systems), where engineers design and build systems with physical dimensions of micrometers. These miniature devices require multiphysics design and simulation tools because virtually all MEMS devices involve combinations of electrical, mechanical, and fluid-flow phenomena.

Available application modes are:

- Plane stress
- Plane strain
- Axisymmetry, stress-strain
- Piezoelectric modeling in 2D plane stress and plane strain, axisymmetry, and 3D solids.
- 3D solids
- Film damping
- Electrokinetic flow
- General laminar flow, including Stokes flow and multiphase flow

The MEMS module also includes predefined multiphysics couplings for thermal-structural, thermal-electric-structural, acoustic-structural, and fluid-structure interaction.

The MEMS Module Model Library contains a suite of models of MEMS devices such as sensors, actuators, and microfluidics systems. The models demonstrate a variety of multiphysics couplings and techniques for moving boundaries.

This module also provides interfaces for integrating SPICE circuits and importing ECAD drawings.

The RF Module

The RF Module provides a unique environment for the simulation of electromagnetic waves in 2D and 3D. With this module you can run harmonic, transient, and

eigenfrequency simulations in an easy-to-use graphical user interface. For example, use the RF Module to simulate electromagnetic wave propagation in microwave components and photonic devices.

The RF Module is useful for component design in virtually all areas where you find electromagnetic waves, such as:

- Antennas
- Waveguides and cavity resonators in microwave engineering
- Optical fibers
- Photonic waveguides
- Photonic crystals
- Active devices in photonics

The available application modes cover the following types of electromagnetics field simulations:

- In-plane wave propagation
- Axisymmetric wave propagation
- Full 3D vector wave propagation
- Full vector mode analysis in 2D and 3D

Material properties include inhomogeneous and fully anisotropic materials, media with gains or losses, and complex-valued material properties. In addition to the standard postprocessing features, the RF Module supports direct computation of S-parameters and far-field patterns. You can add ports with a wave excitation with specified power level and mode type, and add PMLs (perfectly matched layers) to simulate electromagnetic waves that propagate into an unbounded domain. For time-harmonic simulations, you can use the scattered wave or the total wave. The multiphysics capabilities of COMSOL Multiphysics can couple simulations with heat transfer, structural mechanics, fluid flow formulations, and any physical phenomena.

This module also provides interfaces for integrating SPICE circuits and importing ECAD drawings.

The Structural Mechanics Module

The Structural Mechanics Module solves problems in structural mechanics and solid mechanics, adding special element types—beam, plate, and shell elements—for engineering simplifications.

Available application modes are:

- Plane stress
- Plane strain
- Axisymmetry, stress-strain
- Piezoelectric modeling
- 2D beams, Euler theory
- Thick plates, Mindlin theory
- 3D beams, Euler theory
- 3D solids
- Shells

Supporting both linear and nonlinear material models, the module's analysis capabilities include static, eigenfrequency, transient, frequency response, and parametric analyses, as well as contact and friction.

The Structural Mechanics Module 3.5 provides predefined interfaces for elasto-plastic, hyperelastic, and viscoelastic material models. The underlying equations for structural mechanics are always available—a feature unique to COMSOL Multiphysics. This equation-based modeling concept provides unsurpassed flexibility. The Structural Mechanics Module 3.5 includes interesting models that use this capability to model nonlinear materials such as rubber and viscoplastic materials. Material models can be isotropic, orthotropic, or fully anisotropic, and you can use local coordinate systems to specify material properties. In addition, advanced tools for fatigue analysis are available. The Structural Mechanics Module also features an extensible materials library and a beam cross-section library as well as command-line tools for flexible and extensible fatigue analysis tools from COMSOL Script or MATLAB.

The CAD Import Modules

COMSOL 3.5 includes packages for importing CAD drawings into COMSOL Multiphysics. A base package, the CAD Import Module, provides import of the most popular formats: Parasolid, SAT, STEP, and IGES. It also provides bidirectional interfaces to SolidWorks® and Autodesk Inventor®. In addition, we provide a range of add-on modules that import additional formats.

- CAD Import Module, for importing CAD drawings in Parasolid, SAT (ACIS®), STEP, IGES formats, and bidirectional interfaces to SolidWorks and Autodesk Inventor.
- Pro/E Import Module, which requires the CAD Import Module, imports CAD drawings in Pro/ENGINEER format.
- CATIA V4 Import Module, which requires the CAD Import Module, imports CAD drawings in CATIA V4 format.
- CATIA V5 Import Module, which requires the CAD Import Module, imports CAD drawings in CATIA V5 format.
- Inventor Import Module, which requires the CAD Import Module, imports CAD drawings in Autodesk Inventor format.
- VDA-FS Import Module, which requires the CAD Import Module, imports CAD drawings in VDA-FS format.

Internet Resources

A number of Internet resources provide more information about COMSOL Multiphysics, including licensing and technical information. This section provides information about some of the most useful web links and email addresses.

COMSOL Web Sites

Main corporate website: <http://www.comsol.com/>

Worldwide contact information: <http://www.comsol.com/contact/>

Online technical support main page: <http://www.comsol.com/support/>

COMSOL Support Knowledge Base, your first stop for troubleshooting assistance, where you can search for answers to any COMSOL questions:

<http://www.comsol.com/support/knowledgebase/>

Product updates: <http://www.comsol.com/support/updates/> (to check for updates, choose **Check For Updates** from the **Help** menu)

CONTACTING COMSOL BY EMAIL

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

Send your COMSOL technical support questions to support@comsol.com. You will receive an automatic notification and a case number by email.

COMSOL User Forums

On the COMSOL web site, we maintain a list of current COMSOL user forums: <http://www.comsol.com/support/forums/>

These can be Usenet newsgroups or user forums on other websites where COMSOL users share solutions and tips on COMSOL modeling and other issues related to COMSOL products. These user forums are typically independent and unmoderated.

Note: COMSOL's technical staff participates in these user forums as individuals. To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com.

Geometry Modeling and CAD Tools

The CAD tools in COMSOL Multiphysics provide many possibilities to create geometries using solid modeling and boundary modeling. This chapter covers geometry modeling in 1D, 2D, and 3D with examples of solid modeling, boundary modeling, Boolean operators, and other CAD tools in COMSOL Multiphysics. In addition, it shows how to use the tools for exploring geometric properties. This chapter also provides information about using external CAD data and geometries defined using image data.

The COMSOL Multiphysics Geometry and CAD Environment

Overview of Geometry Modeling Concepts

In COMSOL Multiphysics you can use *solid modeling* or *boundary modeling* to create objects in 1D, 2D, and 3D. They can be combined in the same geometry (*hybrid modeling*).

During solid modeling you form a geometry as a combination of solid objects using *Boolean operations* like union, intersection, and difference. Objects formed by combining a collection of existing solids using Boolean operations are known as *composite solid objects*. Boundary modeling is the process of defining a solid in terms of its boundaries. You can combine such a solid with *geometric primitives*—common solid modeling shapes like rectangles, circles, blocks, cones, and spheres, which are directly available in COMSOL Multiphysics.

In 3D, you can form 3D solid objects by defining 2D solids in *work planes* and then *extrude* and *revolve* these into 3D solids. It is also possible to *embed* 2D objects in the 3D geometry.

You can also overlay additional nonsolid objects on top of solid objects to control the distribution of the mesh and to improve postprocessing capabilities. For example, add a curve object to a geometry to control the element size along a 3D curve or add a point to guarantee a mesh vertex in a specific location.

All this means that you can work both with a “top-down” approach starting with geometric primitives and a “bottom-up” approach using boundary modeling.

You can import 2D geometries from DXF files and 3D geometries from STL and VRML files. See “Importing and Exporting Geometry Objects and CAD Models” on page 85 for a description of how to use these CAD file formats. The CAD Import Module provides an interface for import of CAD files on Parasolid, SAT (ACIS), STEP, and IGES formats. In addition, the Pro/E Import Module, CATIA V4 Import Module, CATIA V5 Import Module, Inventor Import Module, and VDA-FS Import Module are available for import of CAD data on the formats specific to those 3D CAD packages and standards. Users that have access to the AC/DC Module, RF Module, or the MEMS Module get a specialized interface for ECAD import. With this interface it is possible to import the file formats GDS, ODB++, and NETEX-G in 2D and 3D.

There is more information about the ECAD Import in the User's Guide of the mentioned modules.

For details on the following features that are available through scripting in COMSOL Script or MATLAB, see the *COMSOL Multiphysics Scripting Guide* and the *COMSOL Multiphysics Reference Guide*:

- Converting images and MRI data into a COMSOL Multiphysics geometry.
- Creating 2D and 3D spline curves and surface interpolation of measured data and insert objects into COMSOL Multiphysics.

Setting up Axes and Grid

In the COMSOL Multiphysics user interface you can set limits for the model axes and adjust the grid lines. The grid and axis settings help you get just the right view to produce a model geometry. To change these settings, use the **Axes/Grid Settings** dialog box that you open from the **Options** menu. You can also set the axis limits with the zoom functions.

AXIS SETTINGS

On the **Axis** page you can set the limits for all axes in the coordinate system.

2D Axis Settings

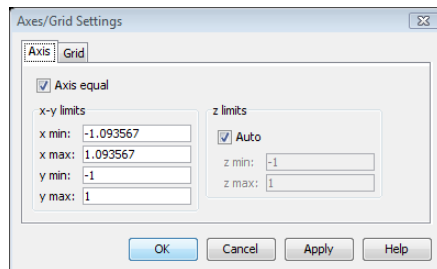


Figure 2-1: The 2D Axes/Grid Settings dialog box.

In 2D, the **Axis equal** option triggers tick mark increments on the *x*- and *y*-axes that are equal in size. Selecting the **Auto** check box makes COMSOL Multiphysics scale the *z*-axis limits automatically. Even though your model geometry does not include the third space dimension, the scale of the *z*-axis can figure into creating height plots, for example, in postprocessing.

3D Axis Settings

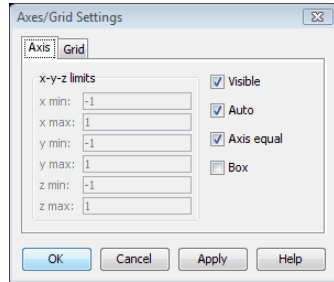


Figure 2-2: The 3D Axes/Grid Settings dialog box.

In 3D, when the **Auto** check box is selected, the axes automatically adjust to enclose the entire geometry. By clearing this check box, you can manually set the axis limits. The **Axis equal** check box, selected by default, scales the data equally. If this feature is turned off, COMSOL Multiphysics scales the data units in a stretch-to-fill fashion. Select the **Box** check box to enclose the axis in a box.

1D Axis Settings

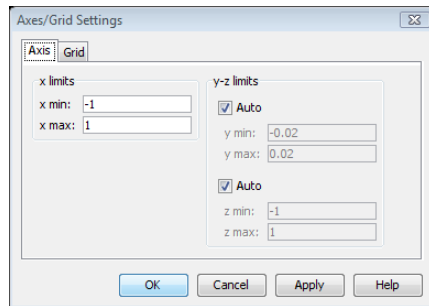


Figure 2-3: The 1D Axes/Grid Settings dialog box.

The **Axis** page in 1D is similar to the 2D version. The difference is that automatic scaling is available for both the *y*-axis and the *z*-axis while equal scaling is not available.

GRID SETTINGS

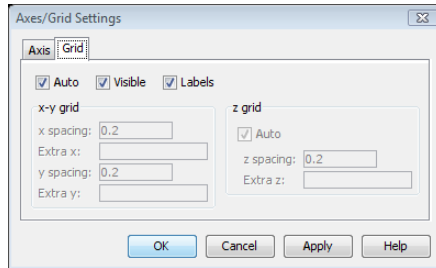


Figure 2-4: Grid settings in the Axes/Grid Settings dialog box.

On the **Grid** page, you set the grid properties. This page has a different set up for 1D and 3D geometries, but the functions are the same for all space dimensions.

Automatic Grid Spacing

Select the **Auto** check box for automatic linear grid spacing.

Manual Grid Spacing

Clear the **Auto** check box to enable edit fields for linear grid spacing and extra tick marks. Then set a linear spacing by entering a scalar spacing. Note that grid spacing always starts at 0. Add extra ticks for drawing the geometry using spaces, commas, or semicolons to separate tick entries.

Examples of extra tick mark locations are:

- π
- $2/3$, 0.78 , 1.1
- -0.123 ; $\pi/4$

Controlling the Grid Appearance

- Select the **Visible** check box to view the grid.
- Select the **Labels** check box to view grid labels.

Creating Cartesian and Cylindrical Coordinate Systems

COMSOL Multiphysics uses a global Cartesian or cylindrical (axisymmetric) coordinate system. You select the geometry dimension and coordinate system when starting a new model in the Model Navigator. By default, variable names for the spatial coordinates are x , y , and z for Cartesian coordinates and r , ϕ , and z for cylindrical

coordinates. These coordinate variables (together with the time for time-dependent models) make up the *independent variables* in COMSOL Multiphysics models.

THE COORDINATE SYSTEMS AND THE SPACE DIMENSION

The default names for coordinate systems vary with the space dimension:

- Models that you open using the space dimensions 1D, 2D, and 3D use the Cartesian coordinates x , y , and z .
- In 1D axisymmetric geometries the default coordinate is r , the radial direction. The x -axis represents r .
- In 2D axisymmetric geometries the x -axis represents r , the radial direction, and the y -axis represents z , the height coordinate.

For axisymmetric cases the geometry model must fall in the half plane $r \geq 0$.

To select the coordinate system and space dimension, select **1D**, **2D**, **3D**, **Axial symmetry (1D)**, or **Axial symmetry (2D)** from the **Space dimension** list in the Model Navigator. You can do this before starting a new model or by clicking the **Add Geometry** button when creating models with multiple geometries.

Note: Not all application modes are available for all coordinate systems and space dimensions.

For axisymmetric models, a red vertical line indicates the symmetry line $r = 0$ when drawing and preprocessing the model.

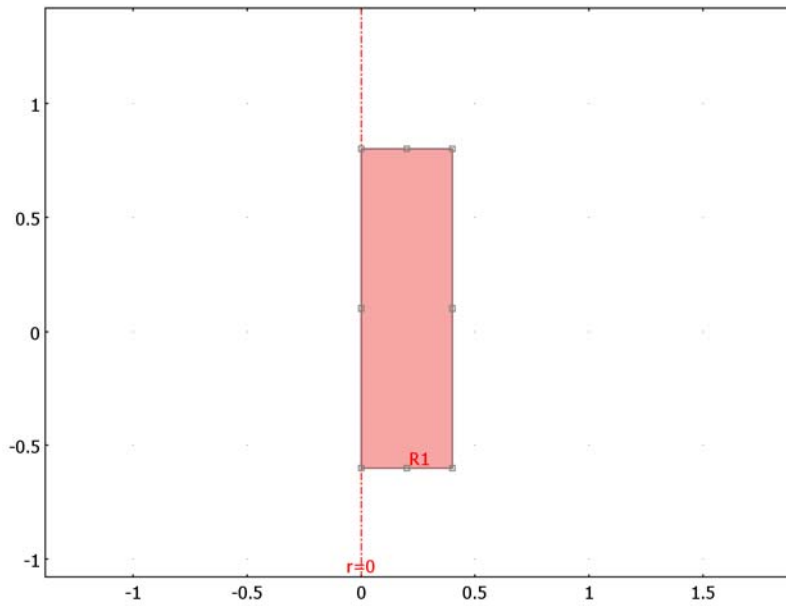


Figure 2-5: The user interface indicates the symmetry line for axisymmetric models. The final geometry must use the half plane $r \geq 0$ only.

Note: The Structural Mechanics Module and MEMS Module use local coordinate systems in addition to the global coordinate system. See the *Structural Mechanics Module User's Guide* or the *MEMS Module User's Guide* for more information on local coordinate systems.

CHANGING THE NAMES OF THE SPACE COORDINATES

The Cartesian coordinates are represented with the space coordinates (independent variables) x , y , and z . The axisymmetric or cylindrical coordinates are r , ϕ , and z . You can change the default names of the space coordinates as when, for example, using a space coordinate to represent time. To change the coordinate names, follow these steps in the Model Navigator:

- 1 Click the **Multiphysics** button.

- 2 Click the **Add Geometry** button.
- 3 Select the space dimension from the **Space dimension** list. This provides the predefined names for the independent variables.
- 4 Replace the predefined names for the space coordinates by entering the names you want in the **Independent variables** edit field. Each name should be separated by a space, and the number of variables must equal the number of space dimensions.

Many 2D application modes use the third space dimension to represent other information, such as time. We recommend, therefore, that you always specify three different space coordinates. Note that if you do not specify a third space coordinate, COMSOL Multiphysics uses the default name (z for Cartesian geometries and phi for axisymmetric geometries).
- 5 Click **OK**.
- 6 Select one or more application modes and click **Add** to add physics or equations to the geometry unless you want to work with just the geometry and the mesh.

Note: Variables that depend on space coordinates, such as vector-field components and space derivatives, use the independent variable names for the geometry where they are defined. For example, if the dependent variable in the model is the pressure p , you access the derivative of pressure with respect to x by typing px in an edit field. These variable names change according to the names you assign to the independent variables. See “Variable Naming Conventions” on page 169 for more information about variable names.

The Status Bar

At the bottom of the COMSOL Multiphysics user interface a status bar shows information and provides buttons for changing some user interface properties. The contents depend on the space dimensions of the current geometry in your model.

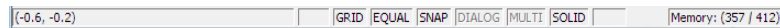


Figure 2-6: The status bar.

Double-clicking a status bar button toggles its features on and off. The status bar components are (from left to right):

- **Information field:** Displays the current cursor position. In 2D Draw mode, this field shows displacements and dimensions of geometric objects when editing and creating those objects. In 3D, it shows the current operation (pan, dolly in/out, and so on) when you change the camera position, including the camera's azimuth and elevation angles when orbiting.
- **AXIS:** Indicates the status of the axis (available in 3D only).
- **GRID:** Indicates the status of the grid lines.
- **EQUAL:** Indicates the status of the *axis equal* feature, where the aspect ratio is set for equal axis lengths (available in 2D and 3D only).
- **SNAP:** Indicates the status of the *snap-to-grid* and *snap-to-vertex* features (available in 1D and 2D Draw mode only).
- **DIALOG:** Indicates if geometry objects are specified using dialog boxes when using Draw toolbar buttons (available in 1D and 2D Draw mode only).
- **MULTI:** Indicates the status of the *draw multiple objects* property that lets you create multiple objects of the same type without having to clicking a Draw toolbar button for each object (available in 1D and 2D Draw mode only).
- **SOLID:** Indicates that geometry objects with closed boundaries are solidified (available in 2D Draw mode only).
- **CSYS:** Indicates the status (visible or hidden) of the 3D coordinate system in the lower left corner of the user interface (available in 3D only).
- **Memory** (in the lower right corner; available on Windows only): Displays the current and peak virtual memory usage in MB, separated by a slash. When the main user interface window is in focus and tooltip display is active, place the cursor over the **Memory** field to see also the current and peak physical memory usage. The physical memory usage includes, as a part, the Java heap, which holds the memory for postprocessing and visualization; see “Rendering and the Java Heap Space” on page 124 for more information on this topic.

Creating Composite Geometry Objects

You can form a *composite geometry object* by combining objects with Boolean operations: set union, set difference, and set intersection. You can form complex geometries using Boolean formulas that include multiple objects and operations. To

perform Boolean operations, use the buttons in the Draw toolbar (2D and 3D) or work in the **Create Composite Object** dialog box.

USING BOOLEAN OPERATIONS FROM THE DRAW TOOLBAR

- Click the **Union** button to create the set union of the selected solid objects.
- Click the **Intersection** button to create the set intersection of the selected objects.
- Click the **Difference** button to create the set difference; this action subtracts the union of all other selected solid objects from the *largest* one.

USING THE CREATE COMPOSITE OBJECT DIALOG BOX

In 2D and 3D you can use the **Create Composite Object** dialog box to create a composite geometry object by combining the selected objects using Boolean operations. Open this dialog box from the **Draw** menu by choosing **Create Composite Object** or by pressing the **Create Composite Object** button on the Draw toolbar.

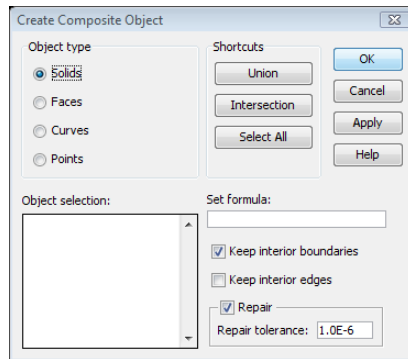


Figure 2-7: The Create Composite Object dialog box.

Using a Set Formula

Choose the solid objects that you want to work with from the list or select them directly in the drawing. Type a set formula that defines the Boolean operation into the **Set formula** edit field. The symbols +, -, and * represent *set union*, *set difference*, and *set intersection*, respectively. You can also use parentheses to form the set formulas. Click **Apply** to form a composite object by combining the selected solid objects according to the set formula. For example, typing $(C1+R1)*E1$ first forms the union of the rectangle R1 and the circle C1 and then forms the intersection of that union and the ellipse E1. Set formulas are only available for solid objects.

Using Shortcut Buttons

The buttons in the **Shortcuts** area assist in forming unions and intersections of all selected objects. Use the **Select All** button to select every geometry object. By default, the set formula represents the union of all selected objects.

Keeping or Removing Interior Boundaries and Edges

Select the **Keep interior boundaries** and **Keep interior edges** (3D only) check boxes to maintain the interior boundaries and edges for the different objects as you combine them. See “Removing Interior Boundaries” on page 35 for more information.

Repair of Composite Objects

When creating a composite object, the software repairs the generated geometry object by removing small edges and faces. To control this feature, open the **Create Composite Object** dialog box, and change the value in the **Repair tolerance** edit field. This value is relative to the overall dimensions of the geometry. For example, if the dimensions are in meters, the default repair tolerance of 10^{-6} makes the geometry repair heal gaps that are smaller than about a micrometer (10^{-6} m). If your geometry has larger defects, you might need to increase this tolerance. To resolve extremely small details, you might need to decrease the repair tolerance. To turn off the repair functionality, clear the **Repair** check box. This is equivalent to using 10^{-10} as repair tolerance. Note that the repair tolerance in **Create Composite Object** is used in all Boolean operations, not only the ones done from this dialog box.

Forming Composite Geometry Objects from Nonsolid Geometry Objects

To create composite objects formed from a combination of solids, faces, curves, and points, you can also use the **Coerce to Solid**, **Coerce to Face**, or **Coerce to Curve** buttons on the Draw toolbar. For more information on coercing geometry objects, see “Coercing Geometry Objects” on page 34.

Moving, Rotating, Scaling, and Mirroring Geometry Objects

Moving, rotating, scaling, and mirroring are *affine transformations* that you can apply to geometry objects. The affine transformations appear on the **Modify** submenu in the **Draw** menu or by clicking the **Move**, **Rotate**, **Scale** and **Mirror** buttons on the Draw toolbar.

- **Move**—Specify displacements in all space directions
- **Rotate**—Specify the rotation angle with a rotation axis or center of rotation
- **Scale**—Specify scaling factors in all space directions
- **Mirror**—Specify the reflection axis or reflection plane

Creating an Array of Geometry Objects

To create an array of identical geometry objects, choose **Modify** in the **Draw** menu and click **Array** or click the **Array** button in the Draw menu. This opens the **Array** dialog box where you can set the displacements and array size.

Copying and Pasting Geometry Objects

Press Ctrl+C or choose **Copy** from the **Edit** menu to copy the selected geometry objects to the COMSOL Multiphysics clipboard.

Press Ctrl+V or choose **Paste** from the **Edit** menu to paste the contents of the COMSOL Multiphysics clipboard into the current model. Use the **Paste** dialog box to specify the displacements of the pasted geometry objects. To paste many copies at the same time, specify several displacements as space-separated displacement coordinates. To paste duplicate geometry objects without using the **Paste** dialog box, choose **Preferences** in the **Options** menu and then clear the **Paste geometry objects using dialog box** check box in the **Preferences** dialog box.

You can also create a copy of a geometry object by pressing the Ctrl key and then drag the copy of the selected object to a new location.

Coercing Geometry Objects

You can transform one or several geometry objects into an object of a different type by *coercing* the object, for example, from a solid to a curve. The resulting object is considered a composite object. The resulting object becomes a solid, face, curve, or point object depending on the target type that you choose in the **Coerce To** submenu in the **Draw** menu. You can also use the **Coerce to Solid**, **Coerce to Face** (3D only), and **Coerce to Curve** buttons as applicable.

The following combinations are meaningful:

2 D

- Coerce a solid object into a curve or point object.
- Coerce a curve object defining at least one closed domain into a solid object.
- Coerce a curve object into a point object.

3 D

- Coerce a solid object into a face, curve, or point object.

- Coerce a face object defining at least one closed domain into a solid object.
- Coerce a face object into a curve, or point object.
- Coerce a curve object into a point object.

Removing Interior Boundaries

Click the **Delete Interior Boundaries** toolbar button or choose the corresponding menu item in the **Draw** menu to delete interior boundaries. In 3D, this also removes edges that are not adjacent to any face and edges adjacent to one face only.

Splitting Geometry Objects

Split a geometry object into its parts by clicking the **Split Object** button on the **Draw** toolbar or by choosing **Split Object** in the **Draw** menu. This splits a solid object with several subdomains into solid objects representing each subdomain. It also splits face, curve, and point objects, which consist of several face, curve, and point segments, respectively, into face, curve, and point objects representing each segment.

Geometry Domain Names

Conceptually, a geometry is a collection of bounded geometric *domains*. The domains are *connected manifolds*, that is, volumes, surfaces, curves, or points. The following table summarizes the technical terms used for these domains:

TABLE 2-1: DOMAIN NAMES IN DIFFERENT SPACE DIMENSIONS

DOMAIN DIMENSION	NAME IN 3D	NAME IN 2D	NAME IN 1D	NAME IN 0D
3	subdomain			
2	boundary	subdomain		
1	edge	boundary	subdomain	
0	vertex	vertex	boundary	subdomain

Domains of maximal dimension are called *subdomains*. Domains of next to highest dimension are called *boundaries*. The boundaries are sometimes referred to as *faces* in 3D and *edges* in 2D. The *vertices* are also called *points*.

The following rules apply to the domains:

- The (interiors of the) domains are disjoint.
- Every domain is bounded by domains of smaller dimension. In particular, a subdomain (in 3D, 2D, or 1D) is bounded by boundaries, edges (in 3D), and

vertices (in 3D and 2D). A boundary (in 3D or 2D) is bounded by edges (in 3D) and vertices. An edge is bounded by vertices.

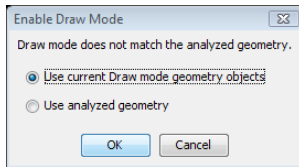
Viewing Geometries

When working with several geometries, you can display the active geometry along with several others. To do so, open the **View Geometries** dialog box from the **Options** menu. In the **Visible geometries** list select the ones to view in addition to the active geometry.

Entering Draw Mode

In certain situations, the Draw mode does not match the current analyzed geometry. This happens, for instance, when you replace the geometry and mesh by a deformed mesh, so that Draw mode still contains the original geometry while the analyzed geometry is the deformed geometry.

If you try to enter Draw mode when it does not match the analyzed geometry, the software displays the **Enable Draw Mode** dialog box where you can choose between keeping the geometry objects in Draw mode or replacing them with the analyzed geometry. When you have made your choice, click **OK** to enter Draw mode.



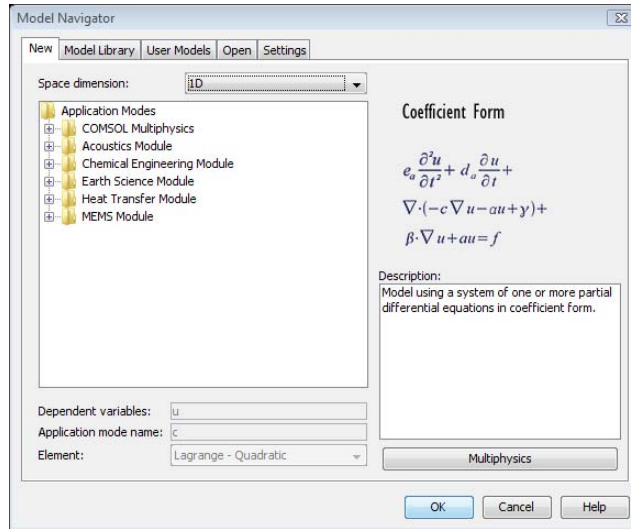
If there is no analyzed geometry, as when working with an imported mesh, the option to use the analyzed geometry is not available. In that case you can use the **Create Geometry From Mesh** dialog box in the **Mesh** menu to create an analyzed geometry and a Draw mode object from the imported mesh.

Note: Opening Draw mode using the **Use current Draw mode geometry objects** option invalidates the old geometry, mesh, and solution.

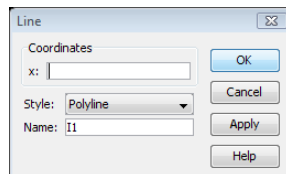
Creating a 1D Geometry Model

To create a 1D geometry model, use the following steps:

- 1 In the **Model Navigator**, click the **New** tab and then select **1D** from the **Space Dimension** list.



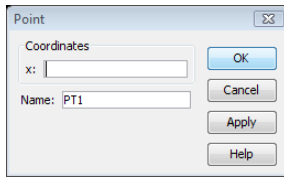
- 2 Click **OK**.
- 3 In Draw mode, open the **Line** dialog box from the **Specify Objects** submenu in the **Draw** menu.



- 4 Enter the coordinates in the **x** field and the name for the object in the **Name** field.
- 5 In the **Style** list you specify the style of the generated solid object. Select **Polyline** to generate a solid object with vertices in the specified coordinates. If you select **Segments**, the specified coordinates must have the structure **start1 end1 start2 end2** and so on. This generates a solid object with segments according to the specified start and end coordinate values.

6 Click **OK**.

Add vertices to the geometric model by choosing **Specify Objects>Point**.



It is also possible to draw solid segments and vertices by clicking the **Line** and **Point** buttons on the Draw toolbar. Then click within the drawing area to specify the vertices. The **Point** dialog box is also available in 2D and 3D.

Creating a 2D Geometry Model

The first step in creating a 2D geometry model is to create a COMSOL Multiphysics model for that space dimension:

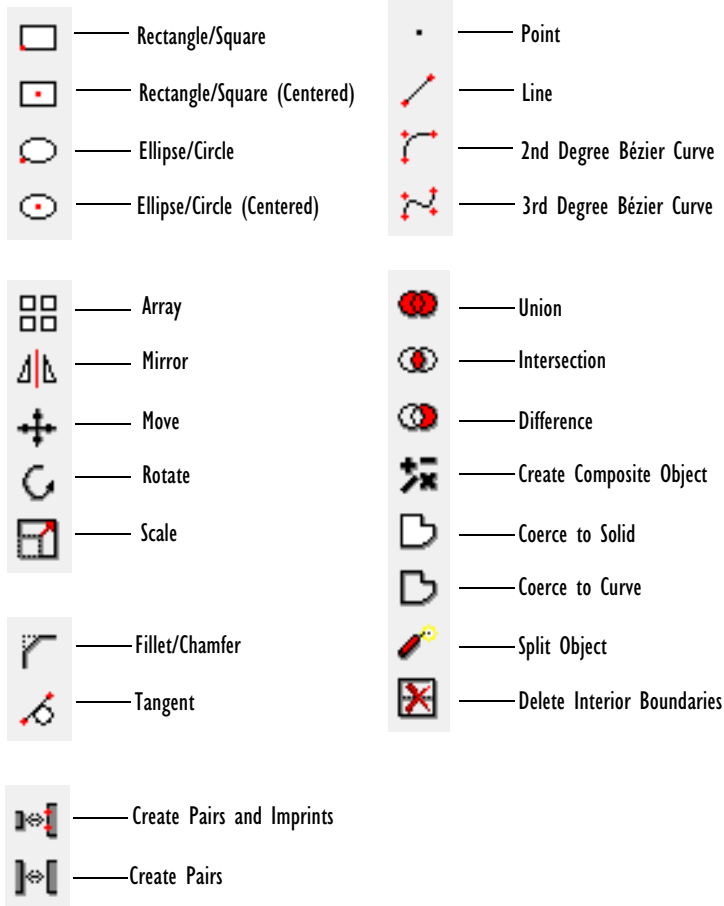
- 1** In the **Model Navigator**, click the **New** tab and then select **2D** in the **Space Dimension** list. Click **OK**.
- 2** Use the CAD tools in the Draw toolbar and the Draw menu to create the 2D geometry. (or import it from a DXF file).

See the following sections on how to use the 2D CAD tools.

Using the 2D Draw Toolbar and the Draw Menu

The default 2D Draw mode provides an additional toolbar for creating and editing geometry objects. This toolbar is also available in the 2D work planes of a 3D model. See the section “Creating and Using 2D Work Planes” on page 59 for details. All drawing tools that are available through the buttons are also accessible from the **Draw**

menu. The 2D Draw toolbar appears to the left of the drawing area. The toolbar consists of five groups of buttons:



2D GEOMETRY OBJECTS

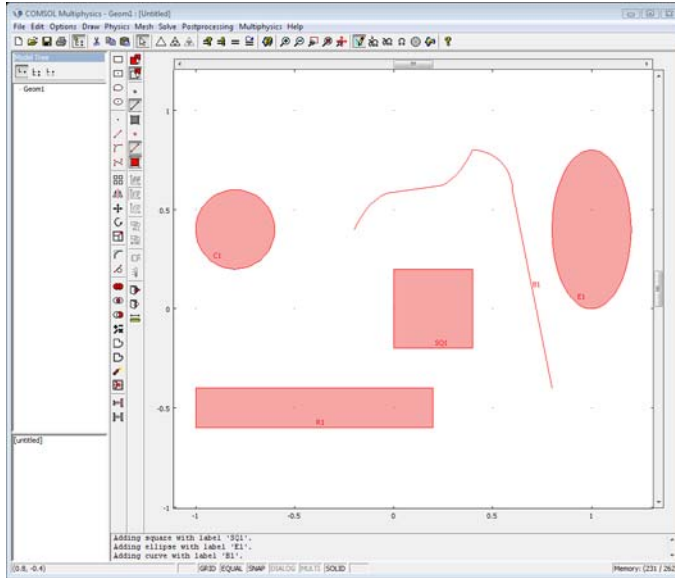


Figure 2-8: Different types of 2D geometry objects.

Using The Draw Toolbar

The Draw toolbar contains tools for creating 2D solid geometric primitives, lines, curves, and points:

- **Rectangle/Square:** Use the left mouse button to create a rectangle; use the right mouse button to create a square, with a reference point at the perimeter.
- **Rectangle/Square (Centered):** Use the left mouse button to create a rectangle; use the right mouse button to create a square, with a reference point at the center.
- **Ellipse/Circle:** Use the left mouse button to create an ellipse; use the right mouse button to create a circle, with a reference point at the perimeter.
- **Ellipse/Circle (Centered):** Use the left mouse button to create an ellipse; use the right mouse button to create a circle, with a reference point at the center.
- **Point:** Click in the drawing to add a point.

For all the above buttons, Shift-clicking the toolbar button opens a dialog box where you can specify the geometry object instead of creating it using the mouse. To make the dialog box appear directly (without Shift-clicking), double-click to select **DIALOG** in the status bar.

- **Line:** Click at the start and end points or drag to create a line.
- **2nd Degree Bézier Curve:** Click at the start, mid, and end points.
- **3rd Degree Bézier Curve:** Click at the start, mid, and end points.

Using the Draw Menu

To creating a 2D geometry object with the mouse: On the **Draw** menu, point to **Draw Objects**, and then select the command for the geometry object that you want to create.

To creating a 2D geometry object using a dialog box to specify location and dimensions: On the **Draw** menu, point to **Specify Objects**, and then select the command for the desired geometry object.

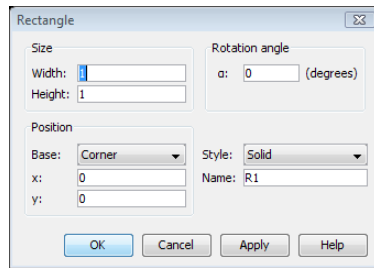


Figure 2-9: The Rectangle dialog box.

In the **Size** area, you specify the size of the geometric primitive (**Width** and **Height** for a rectangle, **Width** for a square, **A-semiaxes** and **B-semiaxes** for an ellipse, and **Radius** for a circle). In the **Position** area you specify the base point of the primitive and if the primitive is centered about that base point or if it has a corner in the base point. Specify the rotational angle (in degrees) about the base point in the **Rotation angle** box. In the **Style** list, select **Solid** or **Curve** to create a primitive of the chosen type. You can also specify the name of the primitive in the **Name** edit field.

For more information on how to create a point and or a line, see “Creating a 1D Geometry Model” on page 37.

2D Solid Modeling Techniques

This section provides a step-by-step example demonstrating how to use Boolean operations to combine solid objects. The Boolean operations can significantly speed up generating complex geometries. The following example shows how to create 2D composite solids starting from primitive solids.

SOLID OBJECTS

A *solid object* consists of an exterior boundary and an interior part. The boundary consists of *edge segments*. For example, a primitive solid rectangle has a boundary consisting of four line segments and one subdomain; a primitive solid circle has a boundary consisting of four circular arcs and one subdomain.

Sometimes a solid object has *interior boundaries* that divide it into more than one subdomain. Consider a square divided into four parts or subdomains. The subdomains can represent regions with different physics or material properties. You can also insert additional points and boundaries to control the mesh inside a solid object.

STARTING A 2D MODEL

Start a new 2D model by going to the **Model Navigator** and selecting **2D** in the **Space dimension** list on the **New** tab. Click **OK**.

CREATING HOLES

First create a rectangle:

- 1 Click the corner-aligned rectangle button at the top of the Draw toolbar; alternately, go to the **Draw** menu, point to **Draw Objects**, and then click **Rectangle/Square**.
- 2 To describe the rectangle's corners, click the left mouse button and drag the cursor from $(-0.8, -0.8)$ to $(0.8, 0.8)$.

Create a circular hole:

- 1 Click the centered ellipse button on the Draw toolbar.
- 2 Click the right mouse button and drag the cursor from $(0, 0)$.
- 3 Release the mouse button when the solid circle has a radius of 0.4.

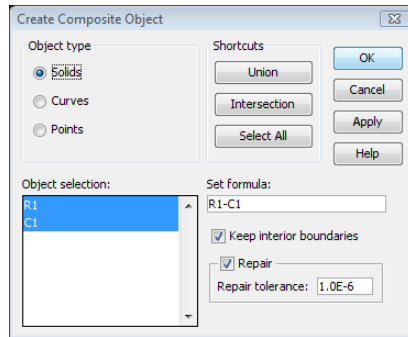
To drill a hole, create a composite solid object:

- 1 Press **Ctrl+A** to select both objects.
- 2 Click the **Difference** button on the Draw toolbar to create a hole.

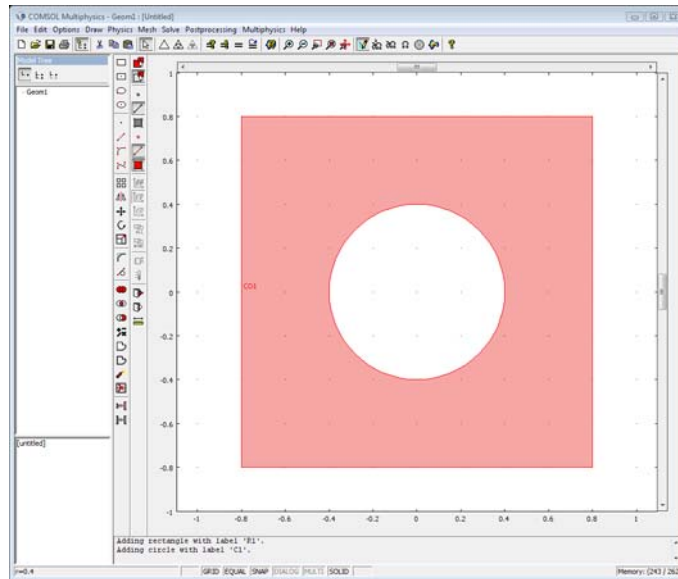
COMSOL Multiphysics offers an alternative approach to drilling a hole:

- 1 Go to the **Draw** menu and choose **Create Composite Object**.
- 2 Click the **Select All** button.

- 3 Edit the **Set formula** by changing the + sign to a – sign so that the set formula becomes $R1 - C1$, subtracting the solid circle from the solid rectangle.



- 4 Click **OK**.

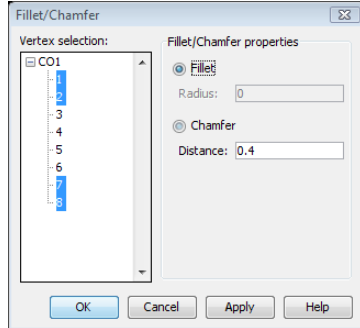


TRIMMING SOLIDS

To flatten the corners on this and other objects, you can use chamfers. To do so, follow these steps:

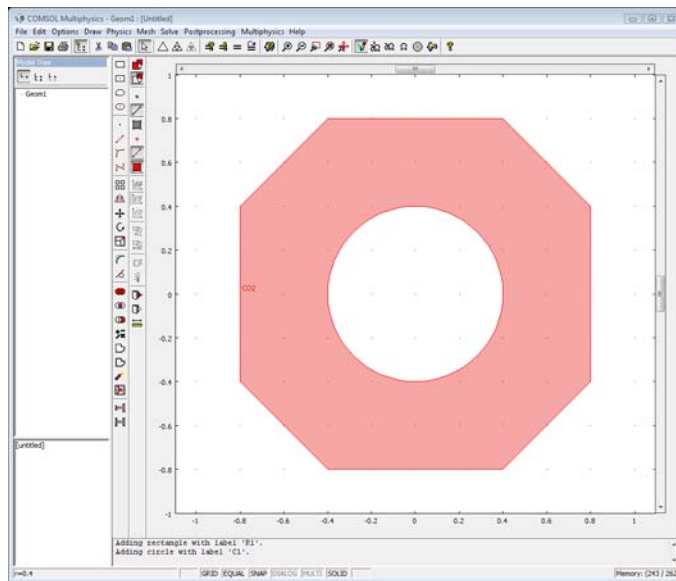
- 1 Click the **Fillet/Chamfer** button on the Draw toolbar to open the **Fillet/Chamfer** dialog box.
- 2 Open the **CO1** folder and select the vertices 1, 2, 7, and 8.

3 Click the **Chamfer** button, and then type 0.4 in the **Distance** edit field.



4 Click **OK** to chamfer the selected corners.

The following figure shows the result.

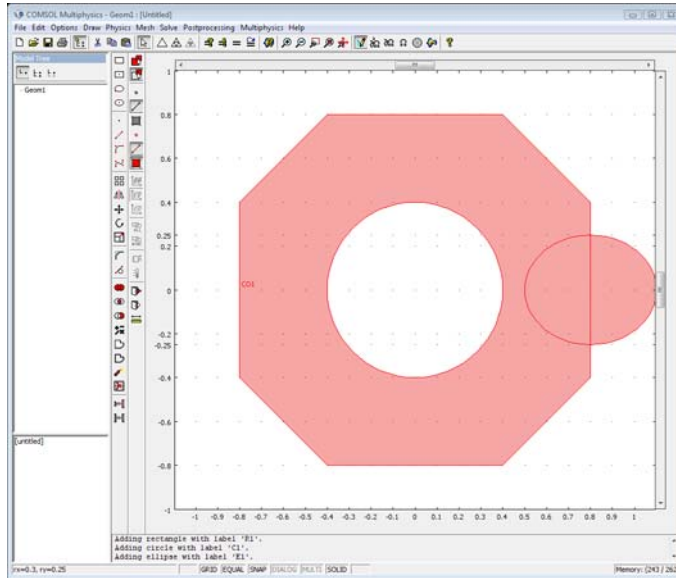


ADDING DOMAINS

To add a domain to a solid, use the union operation. The following example attaches an elliptical domain on the right side of the solid:

- 1 Go to the **Options** menu and open the **Axes/Grid Settings** dialog box.
- 2 On the **Grid** page clear the **Auto** check box.

- 3 Add extra ticks in the y direction by typing `-0.25 0.25` in the **Extra y** edit field.
- 4 Add a grid spacing of `0.1` in the x direction.
- 5 Click **OK**.
- 6 Draw an ellipse centered at $(0.8, 0)$ with semiaxes of length `0.3` and `0.25` in the x and y directions.
- 7 Select all the objects and click the **Union** button on the Draw toolbar.



If you want the combined domain to represent one homogeneous material, click the **Delete Interior Boundaries** button in the Draw toolbar.

2D Boundary Modeling

Boundary modeling explicitly defines a composite solid object's boundary by drawing each curve segment.

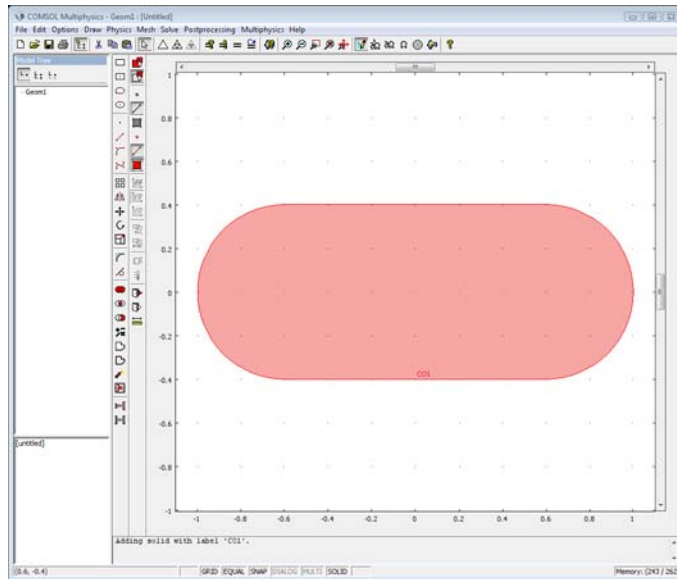
CREATING CURVE OBJECTS

When drawing a curve you create a *curve object*. Primitive curve objects include lines, arcs (2nd degree Bézier curves), and 3rd degree Bézier curves. A composite curve object consists of one or more primitive curve objects. Click the **Coerce to Curve** button in the Draw toolbar to create a composite curve object.

The example below shows how to create a composite solid with boundary modeling. The solid consists of two line segments and four arc segments.

BOUNDARY MODELING USING THE GRAPHICAL USER INTERFACE

- 1 Start a new 2D model.
- 2 Click the **2nd Degree Bézier Curve** button on the Draw toolbar.
- 3 Create the boundary by clicking at $(-0.6, -0.4)$, $(-1, -0.4)$, $(-1, 0)$, $(-1, 0.4)$ and $(-0.6, 0.4)$ with the left mouse button.
- 4 Click the **Line** button on the Draw toolbar.
- 5 Click at $(0.6, 0.4)$ to create a horizontal line segment.
- 6 Once again click the **2nd Degree Bézier Curve** button on the Draw toolbar and then click at $(1, 0.4)$, $(1, 0)$, $(1, -0.4)$ and $(0.6, -0.4)$.
- 7 Click the right mouse button to create a solid object.



CREATING CIRCULAR ARCS

The solid you just created has rounded corners consisting of circular arcs. The shape of the arc depends on the size and shape of the control polygon. COMSOL Multiphysics draws a circular arc only if the control polygon is *isosceles*. For a circular arc, the distance between the first and second control vertices must equal the second and third.

An arc amounts to a 2nd degree Bézier curve with weights that depend on the shape of the control polygon. You can adapt the weight to the control polygon to create elliptical or circular curve segments. For more information on defining curves, see “Conic Sections” on page 515 in the *COMSOL Multiphysics Reference Guide*.

RATIONAL BÉZIER CURVES

This section introduces the properties of rational Bézier curves.

Why Rational Bézier Curves?

When you create a geometry object with a curved boundary, COMSOL Multiphysics represents it in terms of rational Bézier curves. Rational Bézier curves of 2nd degree represent all types of conic sections including circles, ellipses, parabolas, and hyperbolas. COMSOL Multiphysics includes 3rd degree rational Bézier curves to create additional free-form shapes. These include curves with inflection points (S-shaped curves) and advanced fillets.

Control Vertices and Weights

COMSOL Multiphysics defines a rational Bézier curve of degree n in terms of a control polygon consisting of $n+1$ control vertices and $n + 1$ associated control weights. The curve always interpolates the two end points of the control polygon. A change in the control polygon’s shape produces a change in the curve’s appearance. Its shape always mimics that of the control polygon. The higher the curve degree, n , the more complicated the shapes. Increasing the weight pulls the curve toward the corresponding control vertex.

This interaction between the control polygon and the curves makes the rational Bézier curve useful. Below are examples of how alterations in control vertices and weights affect curve shape.

Using the Graphical User Interface

Start by examining an S-shaped 3rd degree rational Bézier curve. To produce this shape, position the control vertices at $\mathbf{b}_0 = (-1, 0)$, $\mathbf{b}_1 = (-0.4, 0.8)$, $\mathbf{b}_2 = (0.6, -0.6)$,

and $\mathbf{b}_3 = (1, 0)$ in a zigzag pattern. COMSOL Multiphysics starts with the control weights $w_0, w_1, w_2,$ and w_3 all set to 1.

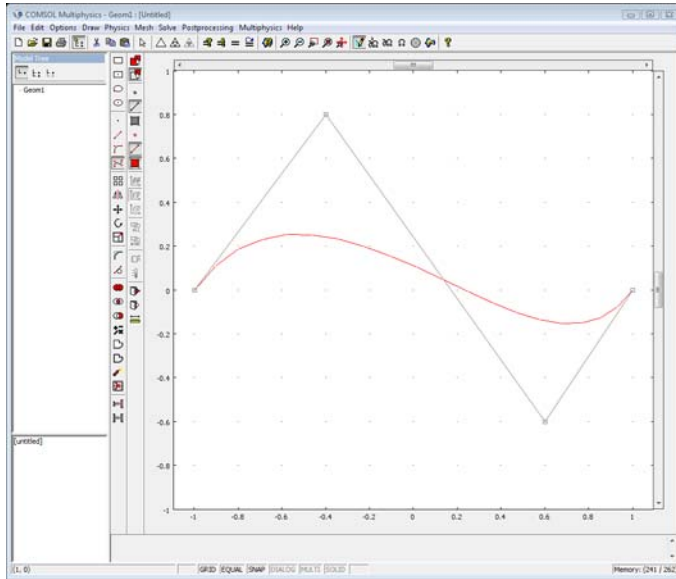


Figure 2-10: A 3rd degree Bézier curve.

Create the curve in Figure 2-10 above:

- 1 Start a new 2D model.
- 2 Double-click **SOLID** on the **Status** bar.
- 3 Click the **3rd Degree Bézier Curve** button on the Draw toolbar.
- 4 Use the left mouse button and click at $(-1, 0)$, $(-0.4, 0.8)$, $(0.6, -0.6)$, and $(1, 0)$.
- 5 Deselect the 3rd degree curve mode by clicking the toolbar button for 3rd-degree Bézier curves.

Creating Fillets and Chamfers

It is possible to modify a corner on a 2D geometry object by creating a fillet (rounding the corner) or a chamfer (flattening the corner). You can perform these operations only on vertices that are adjacent to two boundary segments. Specify the appropriate

properties in the **Fillet/Chamfer** dialog box, which you access using the **Fillet/Chamfer** button on the Draw toolbar or directly from the **Draw** menu.

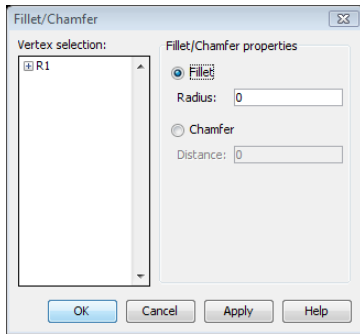
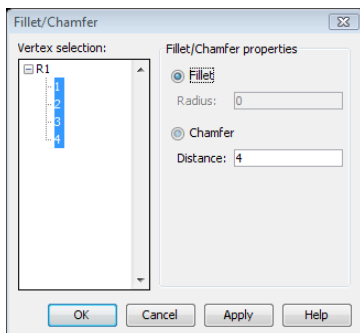


Figure 2-11: The *Fillet/Chamfer* dialog box.

The dialog box contains the **Vertex Selection** list, which gives the names of all vertices on the selected geometry object. After selecting a vertex, go to the **Fillet/Chamfer properties** box and click either the **Fillet** or the **Chamfer** button. For a fillet you also include a value for the fillet radius in the **Radius** edit field; for a chamfer you supply the distance from the corner in the **Distance** edit field.

EXAMPLE

- 1 Start a new 2D model.
- 2 Draw a rectangle with corners at $(-1, 0.8)$ and $(1, 0.8)$.
- 3 Go to the **Draw** menu and open the **Fillet/Chamfer** dialog box.
- 4 Open the **RI** folder to expand the vertex selection list.
- 5 Select all vertices by pressing Ctrl+A.
- 6 Click the **Chamfer** option button, then type 0.4 in the **Distance** edit field.

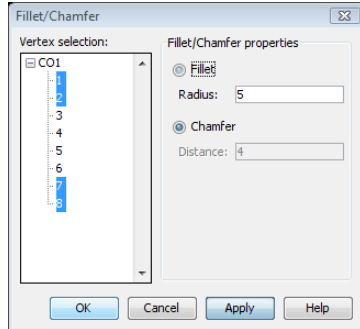


7 Click **Apply** to chamfer all corners.

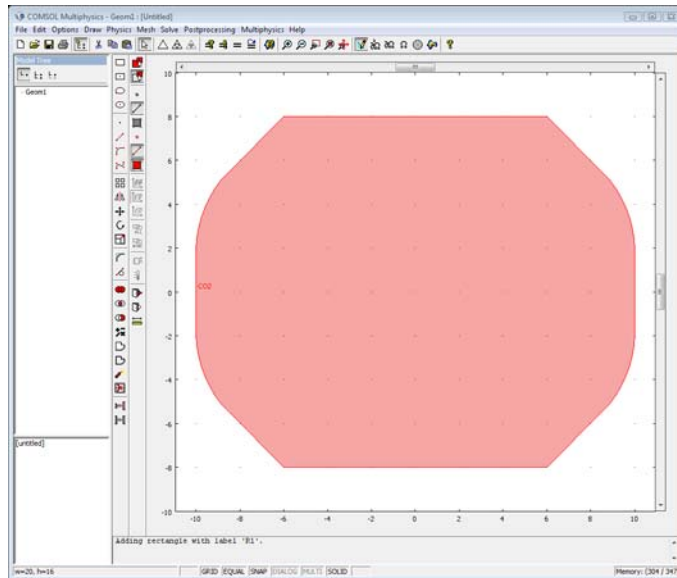
Note that the geometry object changes automatically from a rectangle to a composite object.

8 Open the **CO1** folder and select the vertices 1, 2, 7, and 8.

9 Click the **Fillet** button and then type 0.5 in the **Radius** edit field.



10 Click **OK** to fillet the selected corners and close the dialog box.



Drawing Tangents

Use the **Tangent** dialog box to draw tangents to 2D geometry objects. To open the dialog box, click the **Tangent** button on the Draw toolbar or choose **Tangent** from the **Draw** menu.

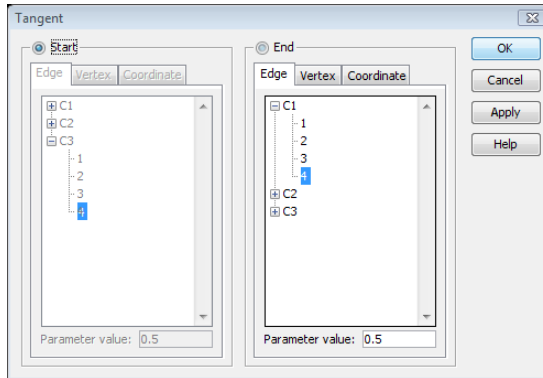


Figure 2-12: The Tangent dialog box.

You can draw tangents between two edges or between an edge and a vertex or point. To create a tangent between two edges, start by selecting the first edge. To select domains, use the list displaying all available domains or click on the domains in the drawing area. To specify the end point, click the **End** button, select the other edge, and then click **Apply** to create the tangent. You can also switch between selecting start or end point by clicking anywhere in the drawing area with the right mouse button. To instead begin or end with a vertex or a point, select the **Vertex** or **Point** tab under the corresponding radio button.

A value between 0 and 1 in the **Parameter value** edit field specifies where on the edge to start searching for a tangent. If you select an edge by clicking on it, the parameter value corresponding to where you clicked automatically appears in this edit field.

If COMSOL Multiphysics cannot find a tangent or if the created tangent is not the intended one, try with different parameter values or select adjacent edges.

EXAMPLE OF USING TANGENTS IN A 2D GEOMETRY

- 1 Start a new 2D model.
- 2 Open the **Circle** dialog box from the **Specify Objects** submenu in the **Draw** menu and create three circles with centers at $(0, 0)$, $(0, 9)$, and $(-6, -4)$ and radii 3, 1.5, and 1.5, respectively.

- 3 Click the **Zoom Extents** button to show the full geometry.
- 4 Go to the **Draw** menu and open the **Tangent** dialog box.
- 5 In the left list, select object C3 to see the available edges and choose edge 4. Alternatively you can select this edge by clicking on it in the drawing area.
- 6 Click the **End** button or right-click anywhere in the drawing area.
- 7 In the right list, select object C1 and choose edge 4. Alternatively you can select this edge by clicking on it in the drawing area.
- 8 Click **Apply** to create the tangent.
- 9 Repeat steps 4 and 5 to draw the following tangents:
 - From object C1, edge 4 to object C2, edge 1.
 - From object C2, edge 2 to object C1, edge 3.
 - From object C1, edge 2 to object C3, edge 2.
- 10 Close the **Tangent** dialog box.
- 11 Select all geometries by pressing Ctrl+A and click the **Coerce to Solid** button in the Draw toolbar.
- 12 Double-click the object to display the **Object Properties** dialog box.
- 13 Select Curves 3 and 4 and click the **Delete** button. Then close the dialog box.
- 14 Click the **Fillet/Chamfer** toolbar button in the Draw toolbar.
- 15 Select Vertex 9, enter 2 in the **Radius** field, and click **OK**.
- 16 Open the **Circle** dialog box from the **Specify Objects** submenu in the **Draw** menu again and create three circles with centers at $(0, 0)$, $(0, 9)$, and $(-6, -4)$ and radii 2, 1, and 1, respectively.
- 17 Select all geometries by pressing Ctrl+A and click the **Difference** toolbar button in the Draw toolbar.

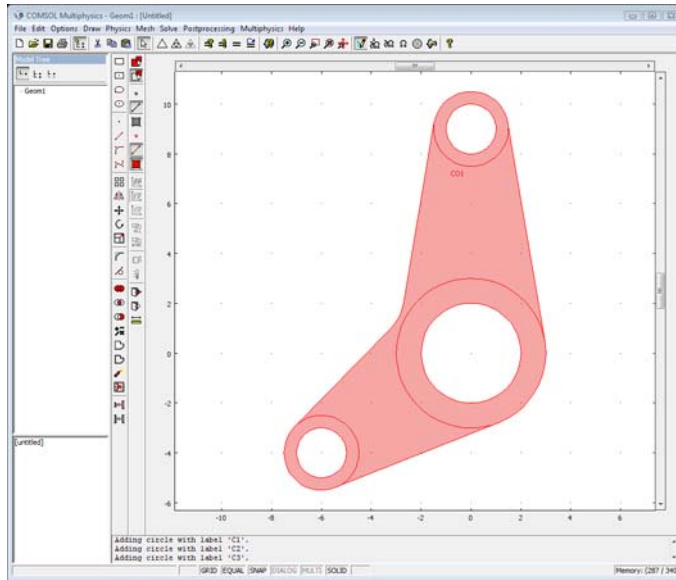


Figure 2-13: The resulting geometry from the previous example.

Editing Geometry Objects

Using the **Object Properties** dialog box, you can edit the properties of a 2D geometry object. Open this dialog box either by double-clicking the geometry object you want to edit or by selecting the object and choosing **Object Properties** from the **Draw** menu.

Opening the dialog box while having selected a 2D primitive (a rectangle, square, ellipse, or circle) brings up that primitive's dialog box (see “Using the 2D Draw Toolbar and the Draw Menu” on page 39 for more information). You can modify the size of a 2D primitive object by clicking on one of its control points and dragging the bounding box with the mouse. To resize an object and retain its original aspect ratio, press the Ctrl key and drag the mouse.

Opening the **Object Properties** dialog box for a nonprimitive instead opens up a dialog box with **Points** and **Curves** pages.

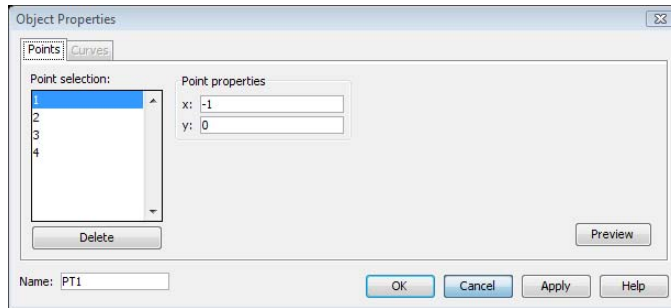


Figure 2-14: The point settings in the Object Properties dialog box.

On the **Points** page you edit coordinates of isolated vertices (points), that is, those not connected to any curve segment. To delete an isolated vertex, select it and click **Delete**. You can also select and move isolated vertices using the mouse while the **Object Properties** dialog box is active.

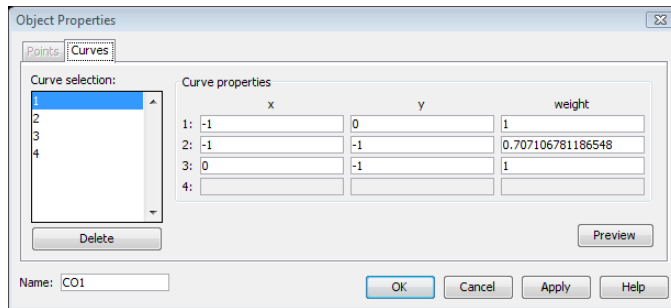


Figure 2-15: The curve segment settings in the Object Properties dialog box.

On the **Curves** page you modify a curve segment's shape and position by editing the positions and weights of the control vertices for the corresponding rational Bézier curve. For more information on this operation see “Rational Bézier Surfaces” on page 516 in the *COMSOL Multiphysics Reference Guide*. To delete a curve segment, select it and click **Delete**. You can also select curve segments and move control vertices using the mouse while the **Object Properties** dialog box is open. Control vertices of a curve segment become visible when a single curve segment is selected.

To see the effect of the changes, click the **Preview** button.

Creating a 3D Geometry Model

The first step in creating a 3D geometry model is to create a COMSOL Multiphysics model with that space dimension:

- 1 In the Model Navigator, click the **New** tab and then select **3D** from the **Space Dimension** list. Click **OK**.
- 2 Use the tools in the Draw toolbar and the Draw menu to create the 3D geometry.

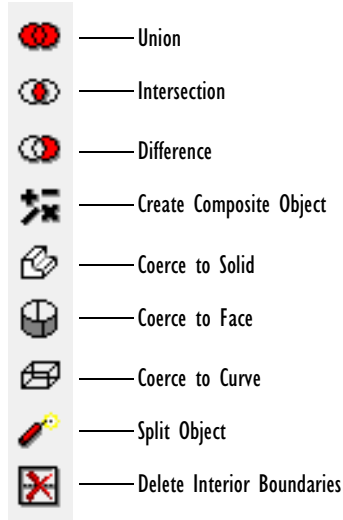
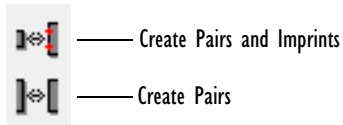
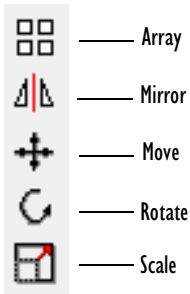
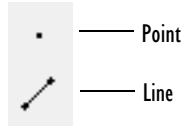
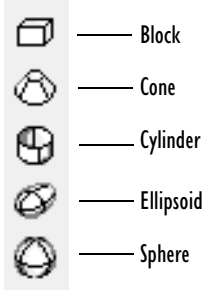
You can also start with a 2D geometry that acts as a work plane and then, for example, extrude the geometry into 3D space. This is often a better approach because you can use the 2D primitives, Boolean operators, and snapping tools before moving into 3D.

See the following sections on how to use the tools for creating 3D geometries.

Using the 3D Draw Toolbar and Draw Menu

The 3D Draw toolbar normally appears to the left of the drawing area. In the work planes an extended 2D Draw toolbar replaces the 3D Draw toolbar. You can access the drawing tools available via these buttons from the **Draw** menu as well. The toolbar contains four groups of buttons:

- Tools for drawing geometry primitives: solids, lines, and points
- Linear transformations
- Set operations and coercions
- Pair tools



SOLID, LINE, AND POINT PRIMITIVES

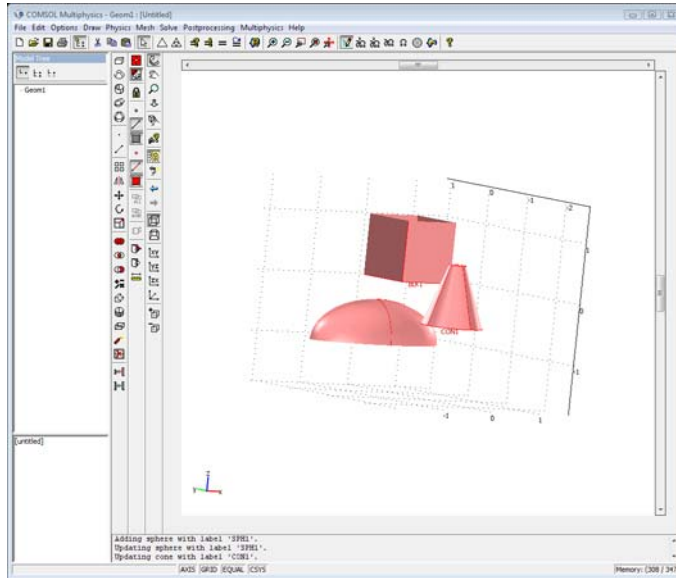
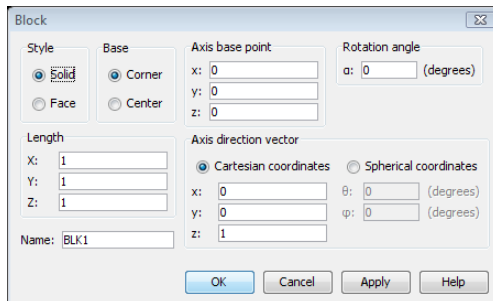


Figure 2-16: Examples of 3D geometric primitives.

Using the Draw Toolbar

On the Draw toolbar you find buttons for creating the following 3D geometric primitives: *block*, *cone*, *cylinder*, *ellipsoid*, and *sphere*. Clicking one of these buttons opens a dialog box for the corresponding primitive.



In the **Block** dialog box, for example, you use the **Style** options to specify if the primitive should be a solid or a face. The size is defined in the **Length** area. In the **Base** area, you choose if the primitive is centered about the base point of the local z -axis or if the primitive has one corner in this base point. To position the base point of the local z -axis

edit values in the **Axis base point** area. To specify the direction of the local z -axis that defines how the primitive is oriented edit the values in the **Axis direction vector** area. The rotational angle about the local z -axis is set in the **Rotational angle** area. You can also specify the name of the primitive object in the **Name** edit field.

The dialog boxes for the other 3D primitives work in a similar way.

For more information on how to create a point or a line, see “Creating a 1D Geometry Model” on page 37.

Using the Draw Menu

Choose the command for the solid you want to create. The commands and the corresponding toolbar buttons open the same dialog boxes to create 3D solids, lines, and points.

Creating 3D Geometries from 2D Geometric Objects

A common way of creating 3D geometries is to model a 2D cross section and create 3D objects using *extrude* and *revolve* operations. You can also *embed* 2D geometries into the 3D geometry model. In addition, it is possible to use Boolean operations and solid modeling to create more complex geometries. In COMSOL Multiphysics you create 2D cross sections in local 2D coordinate systems called *work planes*, which can be positioned anywhere in 3D space. An advantage with this approach is that you can use the snap-to-grid and snap-to-vertex features to connect different geometric parts together. Using numeric values to position geometric objects in 3D is usually not as reliable.

Creating and Using 2D Work Planes

A work plane is a 2D plane positioned in 3D space. Three points and a normal direction define a work plane. You can specify a work plane’s orientation in several ways by:

- Specifying properties explicitly
- Specifying its coordinates
- Using existing 3D geometry objects as reference points

To create a work plane, open the **Work-Plane Settings** dialog box from the **Draw** menu. There are five ways to define a work plane, corresponding to each of the tabbed pages: **Quick**, **Face Parallel**, **Edge Angle**, **Vertices**, and **Advanced**:

QUICK POSITIONING

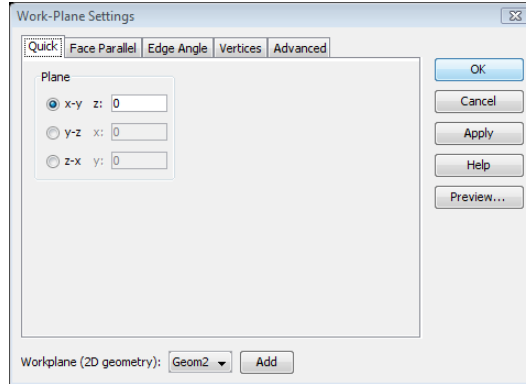


Figure 2-17: Quick positioning settings in the Work-Plane Settings dialog box.

Click the **Quick** tab to create work planes that are parallel to the xy -, yz -, and zx -planes, with an optional offset in the z , x , and y direction, respectively.

FACE PARALLEL POSITIONING

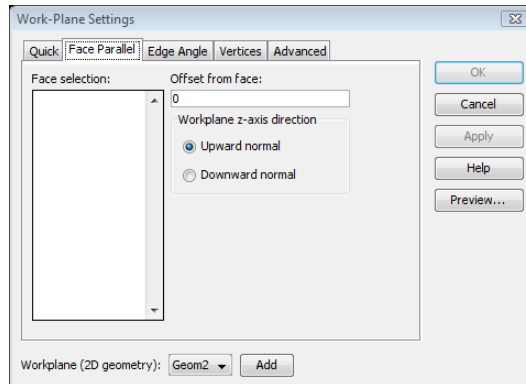


Figure 2-18: Face parallel positioning settings in the Work-Plane Settings dialog box.

Click the **Face Parallel** tab to create a work plane that is parallel to an existing face. Select a face directly from the drawing area or in the **Face selection** list. You can also specify the z -axis direction for the work plane (see “Z-Axis Direction for the Work Plane” on page 62) and an offset from the face.

EDGE ANGLE POSITIONING

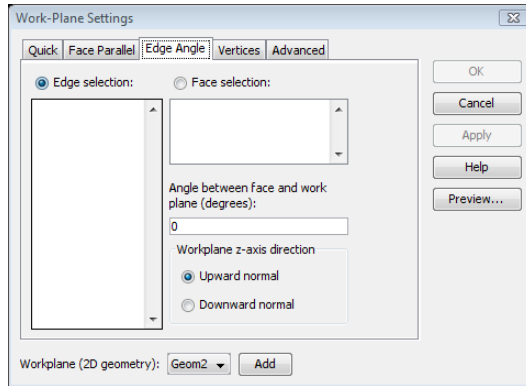


Figure 2-19: Edge angle positioning settings in the Work-Plane Settings dialog box.

Click the **Edge Angle** tab to create a work plane with a specified angle to a selected face. First select an edge to rotate, then select an adjacent face and specify an angle between the work plane and the face.

VERTICES POSITIONING

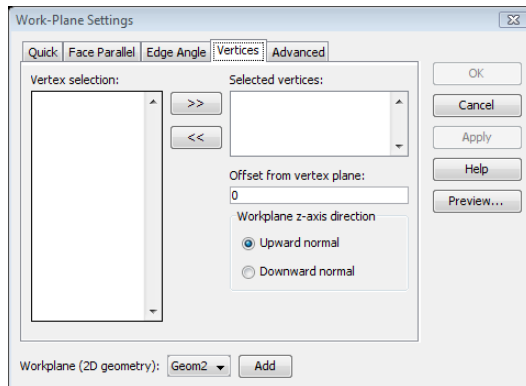


Figure 2-20: Vertices positioning settings in the Work-Plane Settings dialog box.

Click the **Vertices** tab to specify a work plane parallel to another plane specified by three vertices. Select the vertices in the **Vertex selection** list and move them to the **Selected vertices** list by clicking the >> button. You can also specify an offset from the plane. The order of the vertices determines the orientation of the work plane.

ADVANCED POSITIONING

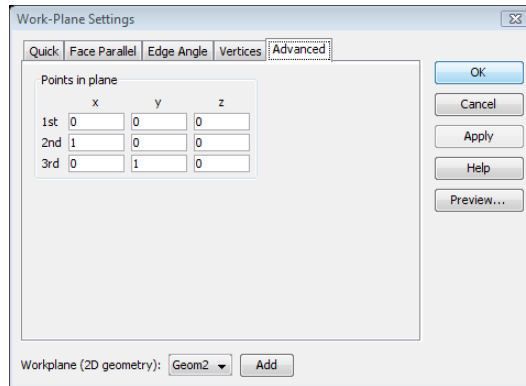


Figure 2-21: Advanced positioning settings in the Work-Plane Settings dialog box.

Click the **Advanced** page to define a work plane using three points in the 3D space. Type the coordinates for the three points in the edit fields in the **Points in plane** area.

Z-Axis Direction for the Work Plane

When defining a work plane parallel to the xy -plane, yz -plane, or zx -plane, using the **Quick** tab, the z -axis direction of the work plane is parallel to z -axis direction, x -axis direction, or y -axis direction of the 3D geometry model, respectively.

For the face parallel, edge angle, and vertices positioning, you can change the z -axis direction of the work plane by selecting the **Upward normal** or **Downward normal** button.

If you click the **Advanced** tab to define the work plane, the z -axis direction of the work plane is defined from the ordering of the specified vertices. The local z -axis direction is parallel to the cross product of the vector starting in the first vertex and ending in the second vertex with the vector starting in the first vertex and ending in the third vertex.

Preview Settings

To help the creation of a work plane, the coordinate system and the geometry in the current work plane are visible while the **Work-Plane Settings** dialog box is open. You can turn off the display of the coordinate system and the work plane geometry in the **Preview Settings** dialog box, which you open by clicking the **Preview** button.

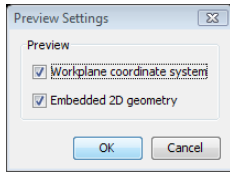
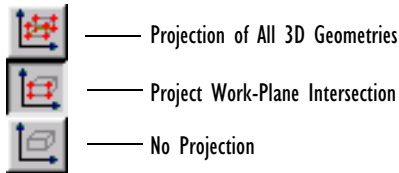


Figure 2-22: The Preview Settings dialog box.

GEOMETRY PROJECTION IN WORK PLANES

When you enter a work plane, a 2D Draw toolbar replaces the 3D Draw toolbar. See “Using the 2D Draw Toolbar and the Draw Menu” on page 39 for details. Three buttons in the Visualization/Selection toolbar determine which portion of the 3D geometry that the software projects onto the plane.



- Click the **Projection of All 3D Geometries** button to draw the projection of all 3D geometry edges
- Click the **Project Work-Plane Intersection** button to display only the edges that lie in the same plane as the work plane
- Click the **No Projection** button to display nothing of the 3D geometry.

Extruding, Revolving, and Embedding

COMSOL Multiphysics provides three ways of extending your 2D work plane geometries into 3D space:

- Extruding
- Revolving
- Embedding

EXTRUDING A 2D GEOMETRY

Choosing **Extrude** from the **Draw** menu opens the **Extrude** dialog box.

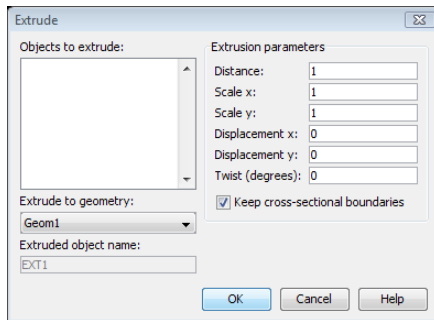


Figure 2-23: The Extrude dialog box.

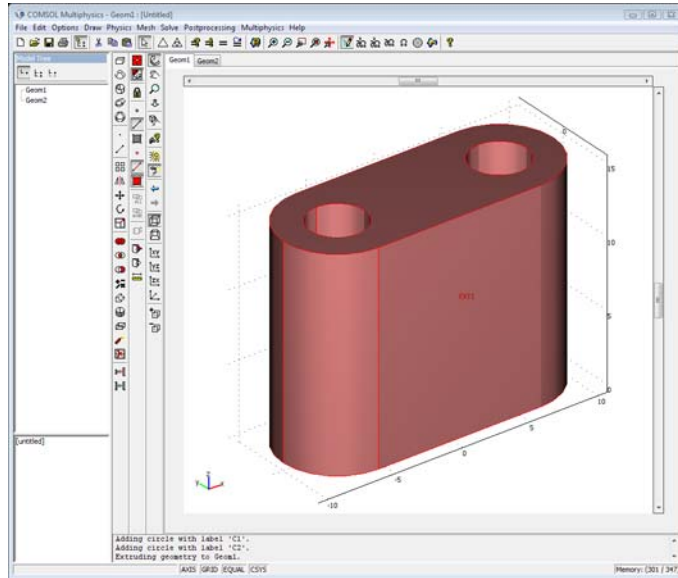
To extrude 2D geometry objects into 3D:

- 1 Select the objects to extrude in the **Objects to extrude** list.
- 2 Specify the length of the extrusion in the **Distance** edit field. To create an extruded geometry with several extrusion sections, type several space-separated values.
- 3 Specify the scale factors in the *x* and *y* directions in the **Scale x** and **Scale y** edit fields. These scale factors determines the difference in size between the end and start of the extrusion. For a straight extrusion, set them to 1.
- 4 Specify the displacements in the *x* and *y* direction at the end of the extrusion in the **Displacement x** and **Displacement y** edit fields. For a straight extrusion, set them to 0.
- 5 Specify the rotation (in degrees) of the top with respect to the bottom of the extruded object in the **Twist (degrees)** edit field.
- 6 Specify if faces should be kept between extrusion sections by selecting the **Keep cross-sectional boundaries** check box. If you clear this check box, the extruded geometry consists of the same number of subdomains as the original 2D geometry.
- 7 Select the 3D geometry to which the resulting 3D geometry object is added in the **Extrude to geometry** list.
- 8 If desired, enter a different name for the extruded object in the **Extruded object name** edit field.
- 9 Click **OK**.

When creating an extruded geometry with several extrusion sections, you can type several space-separated values in each edit field. The number of values in the scale,

displacement, and twist edit fields must be single scalar values or equal to the number of values in the **Distance** edit field.

The figure below shows the result of an extrusion:



REVOLVING A 2D GEOMETRY

Choose **Revolve** from the **Draw** menu to open the **Revolve** dialog box. Here you create 3D objects by rotating the selected 2D geometry objects between two angles and about an axis of rotation:

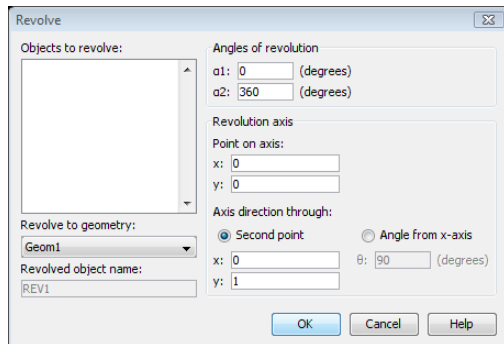
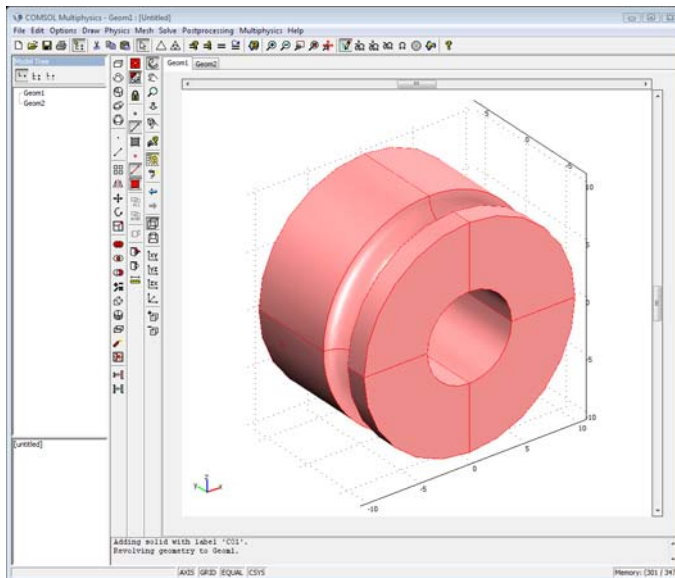


Figure 2-24: The Revolve dialog box.

To revolve 2D geometry objects into 3D:

- 1 Select the objects to revolve in the **Objects to revolve** list.
- 2 In the **Angles of revolution** area, define the angles $\alpha 1$ and $\alpha 2$ (in degrees) for the revolution.
- 3 Use the **Point on axis** edit fields to define a point on the axis of rotation and define axis direction through either a second point or an angle from the x -axis. Click the **Second point** button and specify the coordinates in the x and y edit fields for a second point. Click the **Angle from x-axis** button to enter an angle (in degrees) from the x -axis in the α edit field.
- 4 Select the geometry model to add the resulting 3D geometry to in the **Revolve to geometry** list.
- 5 If desired, enter a different name for the extruded object in the **Revolved object name** edit field.
- 6 Click **OK**.

The figure below shows the result of a revolution.



EMBEDDING A 2D GEOMETRY

From the **Draw** menu, choose **Embed** to open the **Embed** dialog box. Here you can select 2D geometry objects to embed in 3D space. Embedding a 2D geometry object

keeps its 2D properties while including it in the 3D geometry model. You can embed point objects, curve objects, and solid objects. You can also name the embedded object and select to which geometry model the software adds the resulting 3D geometry.

Solid Modeling Using Solid Objects and Boolean Operations

3D SOLID OBJECTS

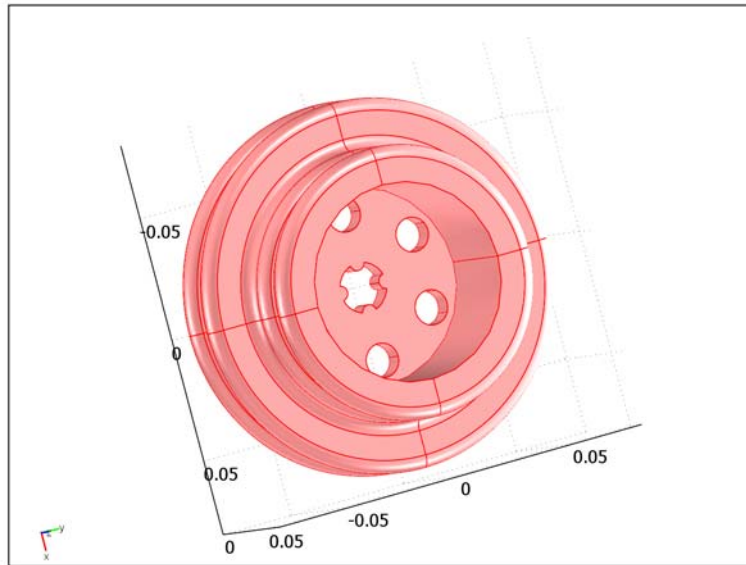
3D solid objects consist of a boundary part and an interior part. The boundary consists of *face segments*. A solid is defined by its exterior and interior boundaries. For example, a primitive solid block has a boundary that consists of six rectangular face segments and one subdomain. A primitive solid cylinder has a boundary consisting of two circular faces, four curved rectangular faces, and one subdomain.

SOLID MODELING AND BOOLEAN OPERATIONS

The following example shows various solid modeling techniques, including the use of work planes and Boolean operations (union, intersection, and difference) to create composite solid objects. The Boolean operations are suited for swift generation of complex geometries. The example is not based on a real component. However, the intention is to cover 3D solid modeling in COMSOL Multiphysics for a geometry that could represent a common mechanical component.

CREATING A 3D PULLEY GEOMETRY

This example creates a 3D pulley using two work planes from which you extrude and revolve 2D cross sections to create the full 3D geometry. Boolean operations, mirroring, rotations, and fillets help to create the 2D cross sections.



The 3D pulley geometry in COMSOL Multiphysics.

Starting a New 3D Model

Open the **Model Navigator** and select **3D** from the **Space dimension** list; then click **OK**.

Creating the Inner Part of the Pulley

Start by creating a 2D solid that you will extrude into 3D to form the inner part of the pulley with the center hole and the surrounding holes. To do so, create a work plane situated in the *xy* plane:

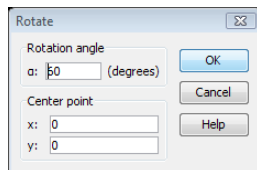
- 1 From the **Draw** menu, open the **Work-Plane Settings** dialog box. Proceed to the **Quick** tab, select the **x-y** button (this is the default setting), and then click **OK**.
You can now draw the cross section of the inner part in this work plane (Geom2).
- 2 Shift-click the **Ellipse/Circle (Centered)** button in the Draw toolbar. This opens the **Circle** dialog box. Type 0.05 in the **Radius** edit field and then click **OK** to create a circle (C1) with a radius of 50 mm centered at the origin.
- 3 Click the **Zoom Extents** button to adjust the axis settings so that they show the full geometry.

Next, create the center hole using four additional circles to cut the shape of the hole:

- 1 Shift-click the **Ellipse/Circle (Centered)** button in the Draw toolbar. This opens the **Circle** dialog box. Type 0.01 in the **Radius** edit field and then click **OK** to create a circle (C2) with a radius of 10 mm centered at the origin.
- 2 Shift-click the **Ellipse/Circle (Centered)** button in the Draw toolbar again. Type 0.003 in the **Radius** edit field and then type 0.01 in the **x** edit field. Click **OK** to create a circle (C3) with a radius of 3 mm centered at the right perimeter of the center circle.
- 3 Press Ctrl+C and then Ctrl+V to make a copy of this circle and paste into the work plane. Type -0.02 in the **x** edit field and click **OK**. This creates a new circle (C4) at the opposite side of the circle center.
- 4 Copy and paste this circle using the same procedure, moving it 0.01 m to the right and -0.01 m down to create a circle (C5) at the bottom perimeter of the center circle.
- 5 Finally copy C5 and paste it with a y-displacement of 0.02 m to create a fourth circle (C6) at the top of the center circle's perimeter.
- 6 Select the circles C2–C6 by Shift-clicking each one. Click the **Difference** button in the Draw toolbar to remove the outer circles. The remaining center geometry is now a composite geometry object CO1.
- 7 Press Ctrl+A to select both objects. Click the **Difference** button in the Draw toolbar to remove the inner part, creating a composite object CO2 with the center hole.

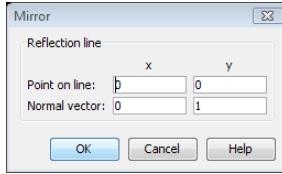
Create six holes with a radius of 7 mm, separated at 60 degree angles around the circle:

- 1 Shift-click the **Ellipse/Circle (Centered)** button in the Draw toolbar. Type 0.007 in the **Radius** edit field and then type 0.03 in the **x** edit field. Click **OK** to create a circle (C1) with a radius of 7 mm centered at the right side of the drawing.
- 2 Press Ctrl+C and then Ctrl+V to make a copy of this circle and paste into the work plane. Click **OK**. This creates a new circle (C2) on top of the first circle.
- 3 Click the **Rotate** button in the Draw toolbar. Type 60 in the α edit field for the rotation angle. Click **OK** to rotate the circle C2 60 degrees counterclockwise.

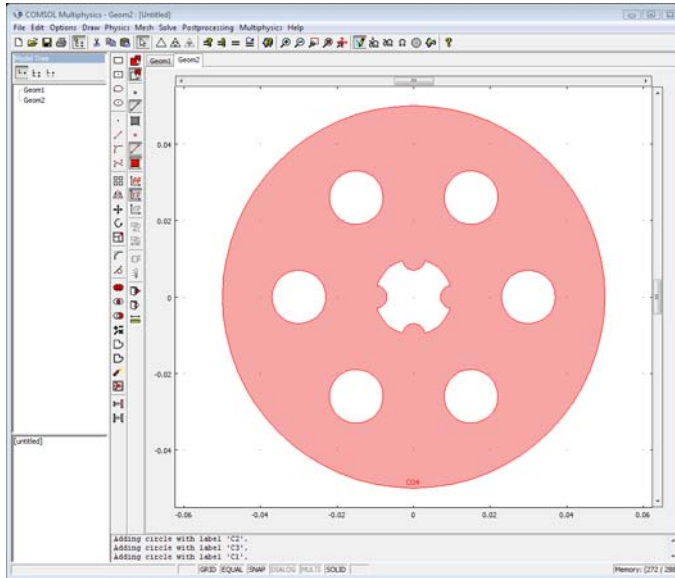


- 4 Repeat Steps 2 and 3 twice to create two more circles, C3 and C4.
- 5 Select the two top circles, C2 and C3.

- 6 Press **Ctrl+C** and then **Ctrl+V** to make a copy of this circle and paste into the work plane. Click **OK**. This creates two new circles, C5 and C6, on top of C3 and C4.
- 7 Click the **Mirror** button in the Draw toolbar. To mirror C5 and C6 about the x -axis, type 0 for the x component and 1 for the y component in the **Normal vector** edit fields. Click **OK**.



- 8 Press **Ctrl+A** to select all objects. Click the **Difference** button in the Draw toolbar to remove the six small circles, creating a composite object CO3 with the center hole and six holes around the center hole.

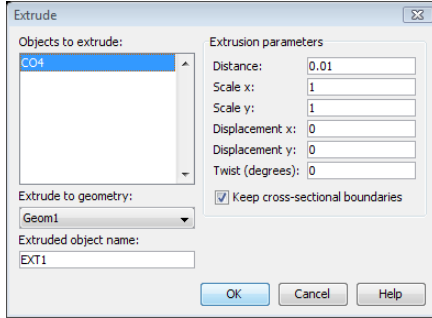


The final geometry in the Geom2 work plane (the xy -plane).

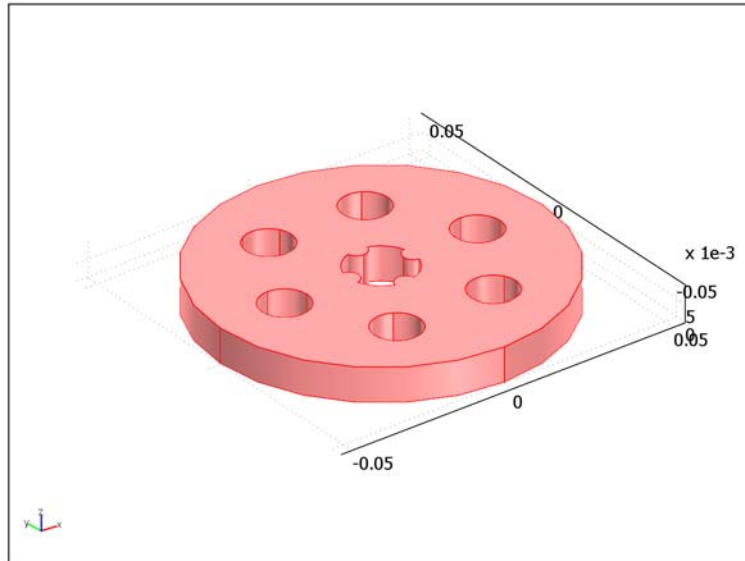
The final step using this work plane is to extrude the 2D geometry into 3D to create a 10 mm thick disk:

- 1 From the **Draw** menu, choose **Extrude**.

2 In the **Extrude** dialog box, type **0.01** in the **Distance** edit field and then click **OK**.



3 In the Camera toolbar, click the **Scene Light** button. The added lighting creates a more realistic 3D image.

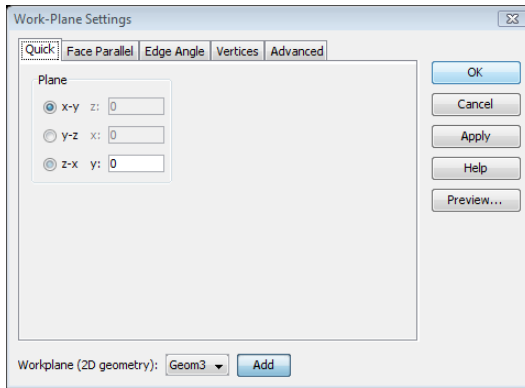


The inner part of the pulley created by extruding the cross section.

Creating the Surrounding Part of the Pulley

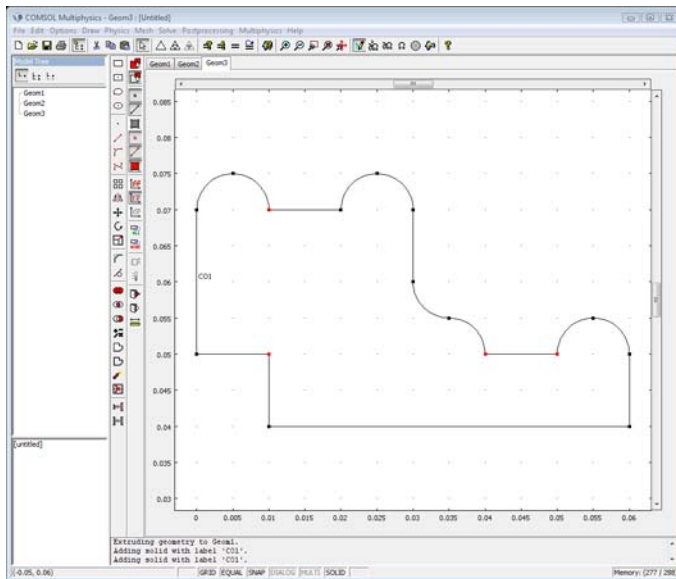
Next, create the surrounding part of the pulley with an inner and an outer track. To do so, create a new work plane where you draw a cross section profile. Then revolve this to form the surrounding part:

- 1 From the **Draw** menu, open the **Work-Plane Settings** dialog box. Click the **Add** button to add a new 2D work plane, Geom 3.
- 2 On the **Quick** page, select the **z-x** button and then click **OK**.



- 3 Click the **Zoom Extents** button to adjust the axis settings so they show the full geometry.
The blue lines that appear indicate the intersection of the 3D disk with the zx plane and help you position the surrounding part. To create the part, first add more grid lines:
- 4 From the **Options** menu, choose **Axes/Grid Settings**.
- 5 In the dialog box, click the **Grid** tab. Clear the **Auto** check box and type **0.005** in the **x spacing** and the **y spacing** edit fields. Click **OK**.
Next, use the line and curve tools to create the boundary of the surrounding part:
- 6 Use the scroll bar to the right of the drawing area to cover the area of the y axis from about **0.03** to **0.08**.
- 7 Click the **Line** button in the Draw toolbar. Click to draw a line from $(0, 0.05)$ to $(0.01, 0.05)$, $(0.01, 0.04)$, $(0.06, 0.04)$, and $(0.06, 0.05)$.
- 8 Click the **2nd Degree Bézier Curve** button in the Draw toolbar. Click at $(0.06, 0.055)$, $(0.055, 0.055)$, $(0.05, 0.055)$, and $(0.05, 0.05)$ to draw a curve segment (half circle).
- 9 Click the **Line** button in the Draw toolbar. Click at $(0.04, 0.05)$ to create a straight line.
- 10 Click the **2nd Degree Bézier Curve** button in the Draw toolbar. Click at $(0.04, 0.055)$, $(0.035, 0.055)$, $(0.03, 0.055)$, and $(0.03, 0.06)$ to draw an S-shaped curve segment.

- 11 Click the **Line** button in the Draw toolbar. Click at (0.03, 0.07) to create a straight line.
- 12 Click the **2nd Degree Bézier Curve** button in the Draw toolbar. Click at (0.03, 0.075), (0.025, 0.075), (0.02, 0.075), and (0.02, 0.07) to draw a curve segment (half circle).
- 13 Click the **Line** button in the Draw toolbar. Click at (0.01, 0.07) to create a straight line.
- 14 Click the **2nd Degree Bézier Curve** button in the Draw toolbar. Click at (0.01, 0.075), (0.005, 0.075), (0, 0.075), and (0, 0.07) to draw a curve segment (half circle).
- 15 Click the **Line** button in the Draw toolbar. Right-click to close the curve and create a 2D solid object (CO1).

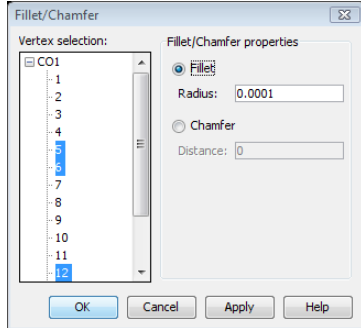


The final cross section. In the next step, you add fillets at the selected vertices.

To remove the sharp angles at the inside of the rounded edges, use fillets with a radius of 1 mm:

- I Click the **Fillet/Chamfer** button in the Draw toolbar.

- 2 Select Vertices (points) 5, 6, 12, and 13. Type 0.001 in the **Radius** edit field.



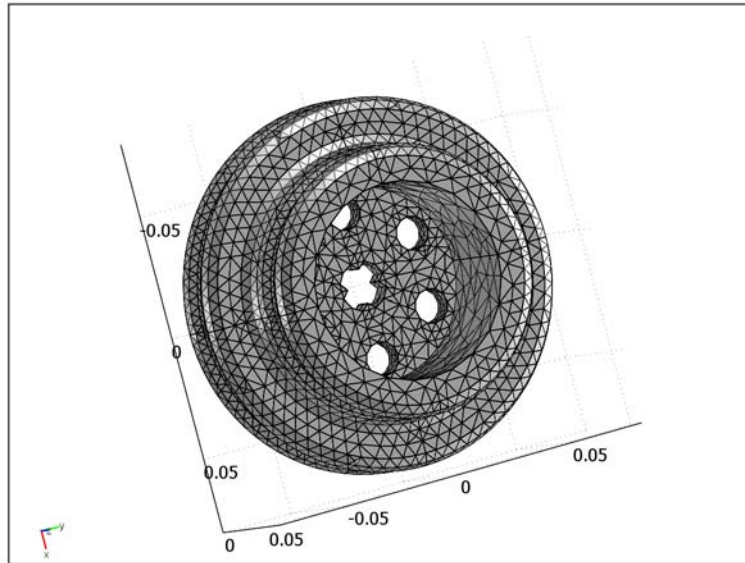
- 3 Click **OK** to create the fillets.

Finally, revolve this cross section a full revolution (360 degrees) about a rotation axis from (0, 0) to (1, 0) in the zx -plane:

- 1 From the **Draw** menu, choose **Revolve**.
- 2 In the **Revolve** dialog box, type 1 in the **x** edit field and 0 in the **y** edit field under the **Second point** button in the **Revolution axis** area.
- 3 Click **OK** to create the 3D object and move to the 3D view.
- 4 Press Ctrl+A to select both 3D objects, click the **Union** button, and then click the **Delete Interior Boundaries** button (both in the Draw toolbar). This makes the entire pulley into one solid object with one single domain. If the parts are made of different materials, skip this step to keep two subdomains in the geometry.

This completes the pulley geometry. Meshing the geometry, you may have to adjust the resolution of narrow regions to avoid an excessive amount of mesh elements

around the filleted areas. The following plot shows an example of a mesh with approximately 80,000 elements:



Editing 3D Objects

Using the **Object Properties** dialog box you can edit a 3D geometry object. Open this dialog box by selecting the object and choosing **Object Properties** from the **Draw** menu.

Opening the dialog box while having selected a 3D geometric primitive brings up that primitive's dialog box (see "Using the 3D Draw Toolbar and Draw Menu" on page 56 for more information).

Opening the **Object Properties** dialog box when you have selected a nonprimitive object instead opens a dialog box where you can delete faces and change the name of the geometry object.

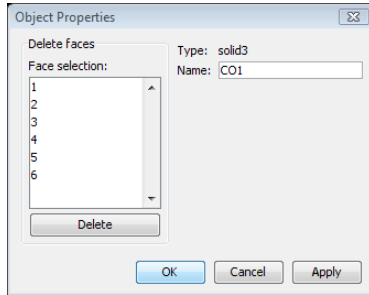


Figure 2-25: The Object Properties dialog box for a 3D solid object.

Exploring Geometric Properties

Introduction

COMSOL Multiphysics includes tools for exploring the following geometric properties of the model geometry:

- Volumes, areas, and lengths
- Surface areas, cross-section areas, and perimeters
- Point coordinates and distance between points
- Center of mass
- Moment of inertia (principal components and tensor components)
- Principal axes for the principal moment of inertia

The center of mass, moment of inertia, and principal axes depend on the density and are available during postprocessing only. The other geometric properties are available also during preprocessing and interactively from the Visualization/Selection toolbar.

You can select and copy the computed values for these geometric properties and then paste them into other settings in a model.

Using The Geometric Properties Dialog Box

You can open the **Geometric Properties** dialog box both from the **Draw** menu and from the **Postprocessing** menu.

GEOMETRIC PROPERTIES DIALOG BOX ON THE DRAW MENU

The **Geometric Properties** dialog box on the **Draw** menu is available in Draw mode and includes the basic properties for the geometry objects in the model. It uses the rendering mesh and does not require a computational mesh or a solution. Use the following steps to display these geometric properties:

- 1 Make sure the user interface is in the Draw mode.
- 2 From the **Draw** menu, choose **Geometric Properties**.
- 3 In 3D, click the **Subdomain** tab to see the volume and surface area for the selected subdomains. The surface area is the area surrounding the selected domains and does not include any interior boundaries or free surfaces.

- 4 Click the **Boundary** tab to see the surface area and perimeter for the selected boundaries. The perimeter is the length of the edges that surround the selected boundaries (faces) and does not include any interior edges or free edges. For example, if you select all boundaries on a cube, the perimeter becomes 0 because all edges are considered interior edges.
- 5 Click the **Edge** tab to see the length of the selected edges.
- 6 Click the **Point** tab to see point coordinates (when you select a single point) and distances between two points (when you select two points).

For planar and axial symmetric 2D geometries, you can display the area and perimeter of subdomains and the length of boundaries. In 1D (planar and axial symmetry), the dialog box shows the length of subdomains. The point coordinates and distance between points are available in 1D, 2D, and 3D.

The dialog box contains a selection list for selecting the objects that you want to compute the geometric properties for. The applicable properties appear in the **Subdomain properties**, **Boundary properties**, **Edge properties**, and **Point properties** areas to the right of the corresponding selection list.

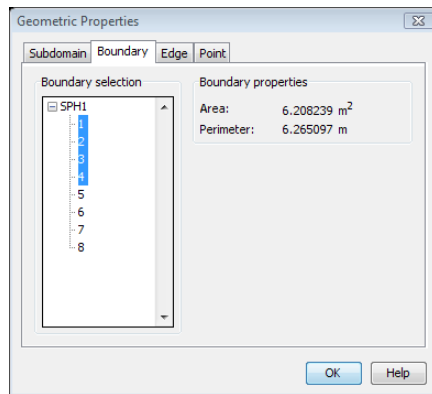


Figure 2-26: The Geometric Properties dialog box opened from the Draw menu.

GEOMETRIC PROPERTIES DIALOG BOX ON THE POSTPROCESSING MENU

The **Geometric Properties** dialog box on the **Postprocessing** menu (see Figure 2-28 on page 81 and Figure 2-29 on page 82) is available in Postprocessing mode and in the Point, Edge, Boundary, and Subdomain selection modes. This version of the dialog box requires that the model contains a solution. You can also make it available without computing a solution. Instead, choose **Get Initial Value** from the **Solve** menu to make

COMSOL Multiphysics use the initial values as the solution for access to the material and geometric properties without a solution.

In addition to the geometric properties in the version on the **Draw** menu (but for the domains in the analyzed geometry instead of the geometry objects), this **Geometric Properties** dialog box also contains the following properties for the domains in the analyzed geometry:

- The center of mass for subdomains (1D, 2D axial symmetry, and 3D), boundaries (2D and 3D), and edges (3D). For 2D axisymmetric models, the center of mass refers to the full 3D geometry and is always located on the z -axis. 1D axisymmetric models do not display the center of mass, which is always at the origin.
- The moment of inertia, which is available in two forms in 2D and 3D:
 - Select **Principal** from the **Moment of inertia** list to display the principal moment of inertia (the mass moment of inertia J_1 , J_2 , and J_3 in 3D; the area moment of inertia I_1 and I_2 , and the mass moment of inertia about a z -axis through the center of mass J_{zz} in 2D), and the corresponding principal axes (\mathbf{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3 in 3D; \mathbf{e}_1 and \mathbf{e}_2 in 2D)
 - Select **Tensor** from the **Moment of inertia** list to display the tensor components for the moment of inertia: the mass moment of inertia J_{xx} , J_{yx} , J_{yy} , J_{zx} , J_{zy} , and J_{zz} in 3D; the area moment of inertia I_{xx} , I_{yx} , and I_{yy} , and the mass moment of inertia about a z -axis through the center of mass J_{zz} in 2D.

In 2D axisymmetric models, the moment of inertia has only one component: J_{zz} , the mass moment of inertia about the z -axis.

The mass moment of inertia has the SI unit $\text{kg}\cdot\text{m}^2$, and the area moment of inertia (second moment of inertia) has the SI unit m^4 . The moment of inertia is not available in 1D.

- For 2D axisymmetric models, the **Geometric Properties** dialog box shows the volume and surface area for the full 3D geometry (the cross section swept 360 degrees about the z -axis) in addition to the area and perimeter of the 2D cross section.
- For 1D axisymmetric models, the **Geometric Properties** dialog box shows the area and volume for the full 2D and 3D geometries (the cross section swept 360 degrees about the z -axis) in addition to the length of the 1D radius. For the volume, COMSOL Multiphysics uses the value in the **Thickness** dialog box.

The center of mass, moment of inertia, and principal axes also require that you provide the density in the **Density expression** edit field. The default value is 1, which means that the values for the center of mass becomes the centroid (center of mass for

homogeneous objects). You can type any number or expression that represents the density (for example, a function of the space coordinates x , y , and z). To use the density from the material properties in the model, type the name of the application mode variable for the density such as `rho_ht` for the density defined in the Heat Transfer by Conduction application mode.

For 2D domains (subdomains in 2D and 1D axial symmetry; boundaries in 3D and 2D axial symmetry), you can also provide a value or expression for the thickness in the **Thickness** edit field. For 1D domains (subdomains in 1D, boundaries in 2D, and edges in 3D), you provide the cross-section area for the transverse cross section in the **Cross-section area** edit field. These values affect the moment of inertia and the principal axes. They also affect the center of mass if the thickness or cross-section area varies.

For models that use more than one frame, such as models with moving meshes based on the ALE method, you can choose which frame you want COMSOL Multiphysics to compute the geometric properties based on. By default, it is the fixed (reference) frame.

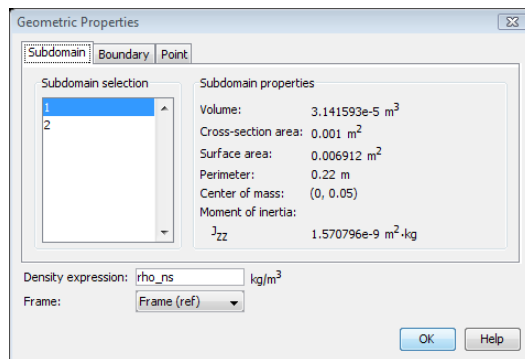


Figure 2-27: The Geometric Properties dialog box opened from the Postprocessing menu for a 2D axisymmetric model: the volume and surface area refer to the full 3D geometry.

Note: The use of the rendering mesh in the **Geometric Properties** dialog box on the **Draw** menu makes the computations of volumes, areas, lengths, surface areas, and perimeters less accurate than the corresponding quantities based on the computational mesh. To obtain the most accurate values for these quantities, use the **Geometric Properties** dialog box on the **Postprocessing** menu instead.

Displaying Geometric Properties in the Message Log

To display basic geometric properties for the selected domains, click the **Geometric Properties** button at the bottom of the Visualization/Selection toolbar. The geometric properties (for example, the volume and surface area for a 3D geometry object or subdomain) then appear in the message log at the bottom of the user interface.



Figure 2-28: The Geometric Properties button on the Visualization/Selection toolbar.

Centroid, Center of Gravity, and Distance in Electronic Conductor

The following example shows how to use the **Geometric Properties** dialog box and button to compute the centroid, center of gravity, and distance between two points in the Electronic Conductor model, which is available in the COMSOL Multiphysics Model Library's Multiphysics folder.

- 1 Open the **Model Navigator**, and click the **Model Library** tab.
- 2 Select **COMSOL Multiphysics>Multiphysics>Electronic Conductor** and click **OK**.
- 3 From the **Postprocessing** menu, choose **Geometric Properties**.
- 4 In the **Geometric Properties** dialog box, click a subdomain in the Subdomain selection list and press Ctrl+A to select all subdomains. This provides a value for the

centroid on the **Center of mass** row, because the default density is 1, which corresponds to a homogeneous material.

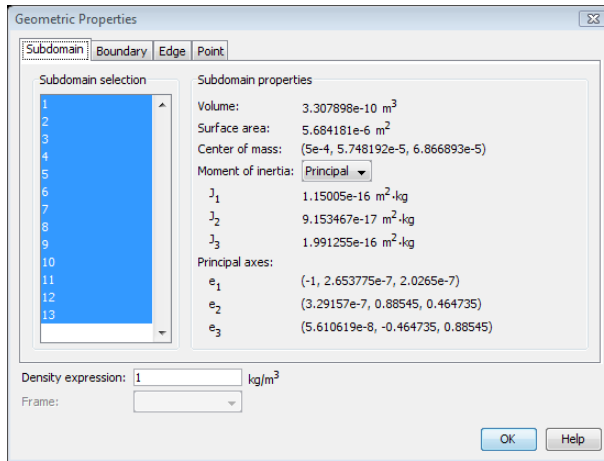


Figure 2-29: The centroid of the electronic conductor geometry (homogeneous density).

- 5 Type `rho_ht` in the **Density expression** edit field and press Return to update the geometric properties. The **Center of mass** row now contains the center of mass. In the following figure, notice that the position moves toward the soldering parts (in the negative y and z directions) because the density of the solder (60Sn-40Pb, 9000 kg/m^3) is higher than the density of the conductor (copper, 8700 kg/m^3).

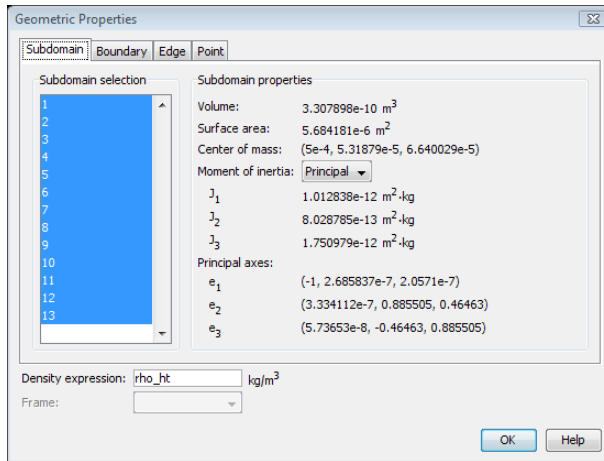


Figure 2-30: The centroid of the electronic conductor geometry (actual density).

- 6 Click **OK**.
- 7 Click the **Point Mode** button on the Main toolbar.
- 8 Click the **Zoom Window** button on the Main toolbar; then click and drag to zoom in the left soldering part.
- 9 Select the two bottom points along the front (Points 7 and 20). Then click the **Geometric Properties** button on the Visualization/Selection toolbar.

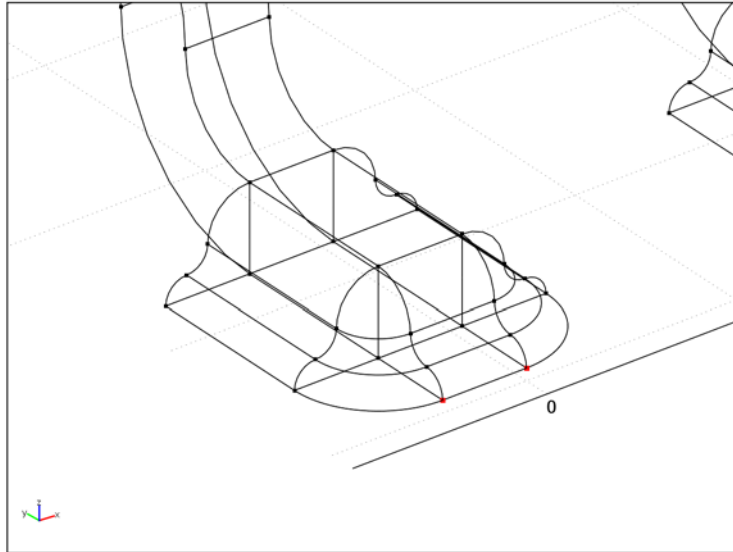


Figure 2-31: Close-up of the left soldering part.

10 COMSOL Multiphysics displays the distance between these points in the message log: $2e-4$ [m] (0.2 mm).

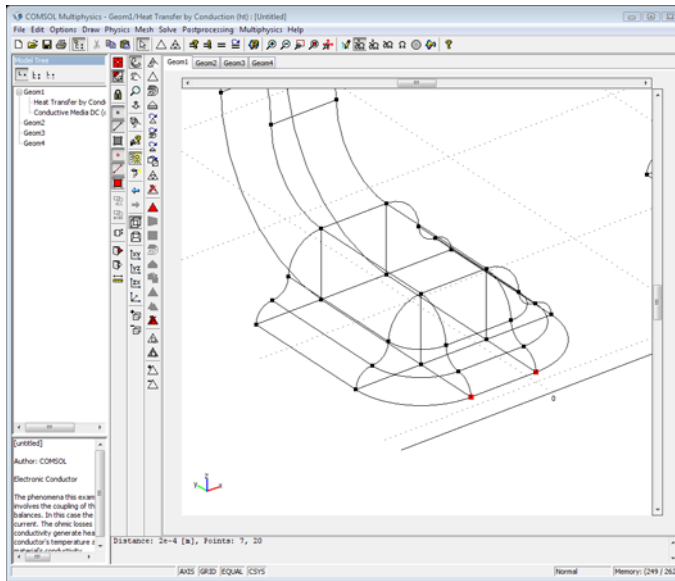


Figure 2-32: The distance between two points appears in the message log at the bottom of the user interface.

Importing and Exporting Geometry Objects and CAD Models

With COMSOL Multiphysics you can import and export geometries from a variety of file formats. A natural choice for storing geometries in 1D, 2D, and 3D is the native file format of COMSOL's geometry kernel. You find more information on this file format in the section “Importing and Exporting COMSOL Multiphysics Files” on page 86.

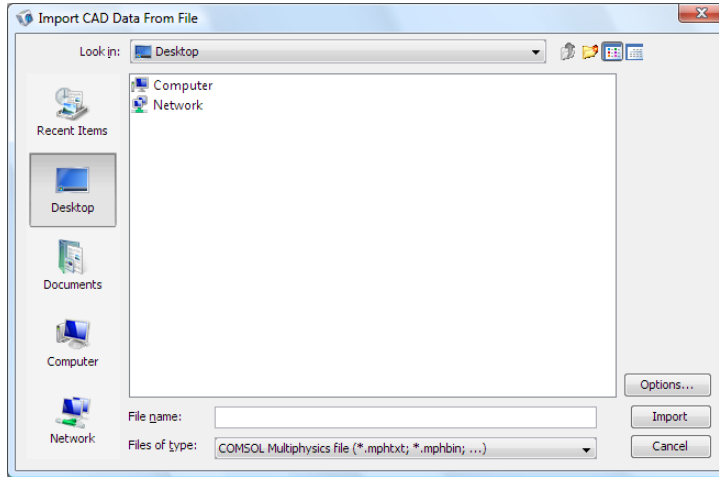
In 2D, there is also the option of importing and exporting DXF files; for details, see the section “Importing and Exporting DXF” on page 87.

In 3D, it is possible to import surface meshes on the STL and VRML formats. For more information about this feature, see the section “Importing STL and VRML Files” on page 89.

COMSOL provides an optional CAD Import Module, which supports import of CAD data on Parasolid, SAT (ACIS), STEP, and IGES formats. Additional modules support the following formats: CATIA V4, CATIA V5, Pro/ENGINEER, Autodesk Inventor, and VDA-FS. The *CAD Import Module User's Guide* covers these file formats in detail.

Importing and Exporting COMSOL Multiphysics Files

You can import geometry objects from COMSOL Multiphysics files by selecting **Import>CAD Data From File** in the **File** menu.



- 1 Choose **COMSOL Multiphysics file** in the **Files of type** list.
- 2 Choose the file to import by clicking on it or specifying its name in the **File name** edit field.
- 3 Click **Import** to close the dialog box and import the geometry in the file.

There is also an **Options** button in the dialog box. The software ignores these options when importing COMSOL Multiphysics files. The CAD Import Modules uses these options when importing CAD files on other formats.

To export geometry objects to a COMSOL Multiphysics file, select **Export>Geometry Objects To File** in the **File** menu. Then choose **COMSOL Multiphysics binary file** or **COMSOL Multiphysics text file** in the **Files of type** list and click **Save**.

Note: COMSOL Multiphysics files (.mphtxt, .mphbin) are not the same as a Model MPH-file (.mph). The former are native formats for the COMSOL geometry kernel and are only used for geometry and mesh objects.

Importing and Exporting DXF

In COMSOL Multiphysics you can import and export 2D CAD drawings on the DXF file format, a common format for 2D CAD drawings. The DXF import is available for 2D geometries and for work planes in a 3D model. The following example describes how to import a 2D DXF file.

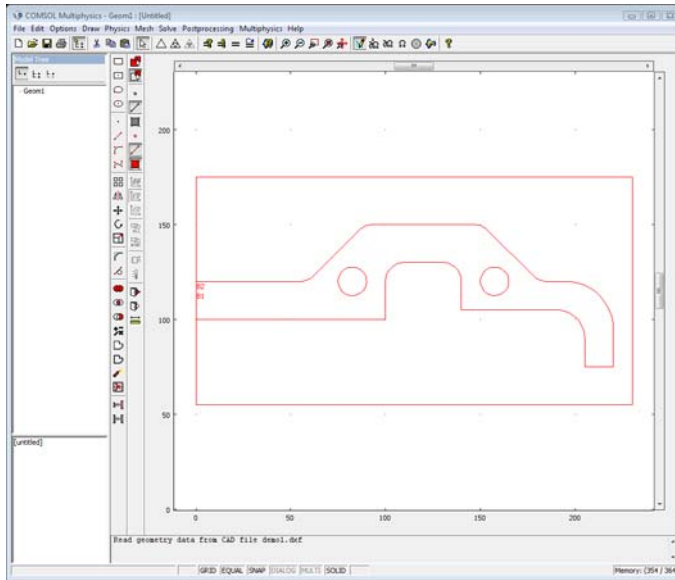
To create a geometry for mesh generation and finite element analysis, COMSOL Multiphysics requires a high degree of accuracy within the CAD drawing. Sometimes the DXF geometries contain small gaps and exceedingly short edges that make it impossible to create a valid 2D solid or a valid mesh. COMSOL Multiphysics provides repair tolerance settings to remove short edges and close small gaps during DXF file import. The repair tolerance in COMSOL Multiphysics specifies the largest distance between the end points of curves allowed in the imported geometry. You can specify this tolerance as an import option (see Figure 2-33 on page 88).

A DXF file contains groups of objects in different layers. The software combines objects within the same layer into one geometry during import. You can specify import options on how to perform this combination (see Figure 2-33 on page 88). The program prints information on imported layers to the log page of the progress window.

Import a DXF file by following these steps:

- 1 Open the **Model Navigator** and start a new 2D model.
- 2 Select the menu item **File>Import>Import CAD Data From File**.
- 3 In the **Files of type** list select **DXF file**.
- 4 Locate the COMSOL installation directory, go to the **demo** folder, then select the file `demo1.dxf`.

- 5 Click **Import** to close the dialog box and import the geometry. This action also opens the **Import CAD Data From File** progress window.



In the **Import CAD Data From File** dialog box, you can click **Options** to open the **CAD Import Options** dialog box.

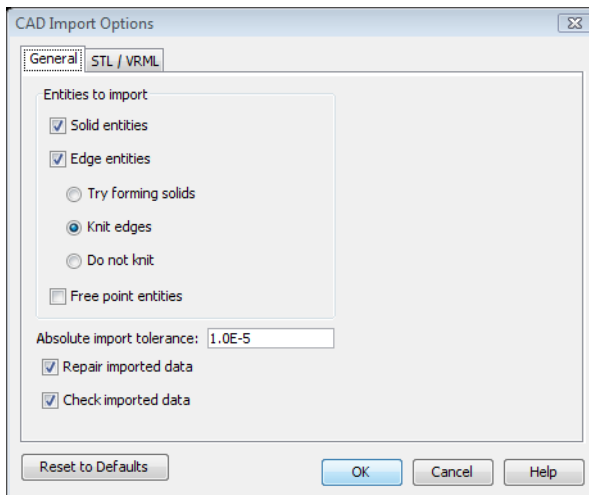


Figure 2-33: CAD Import Options for 2D geometries.

In this dialog box you can control the following options:

- Select the **Solid entities** check box to import solid geometry objects.
- Select the **Edge entities** check box to import curve objects. To choose how to combine these entities:
 - Click the **Try forming solids** button to combine the edges to form solid entities. If there are loose edges not adjacent to a subdomain, they are deleted.
 - Click the **Knit edges** button to combine the edges to a curve object.
 - Click the **Do not knit** button to create a separate geometry object for each edge.
- Select the **Free point entities** check box to import isolated points.
- Select the **Repair imported data** check box to remove small gaps and tiny edges from the geometry. Specify the tolerance in the **Repair tolerance** edit field.

For exporting geometry objects to a DXF file, select **Export>Geometry To File** in the **File** menu. Then, choose **DXF file** in the **Files of type** list and click **Save**.

Importing STL and VRML Files

You can import 3D surface meshes into COMSOL Multiphysics with STL and VRML 1.0, two formats for describing solids as surface meshes.

The STL format describes the boundary of a solid as a triangular mesh. STL files are either ASCII or binary, and COMSOL Multiphysics can import both ASCII and binary STL files. It is advisable to use binary files if possible to avoid a loss of precision.

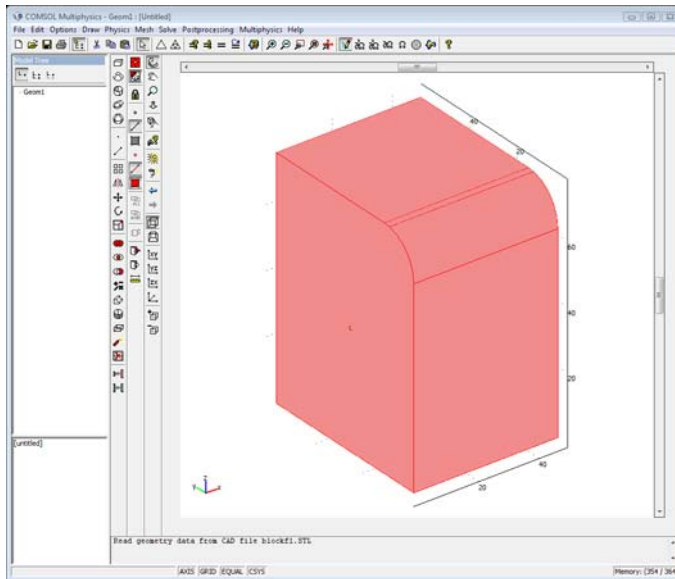
The VRML format describes scenes (collections of independent objects that reside within a single “frame” or logical space), and it can describe the geometry in a scene as a triangular mesh. COMSOL Multiphysics ignores all other information in a VRML file. There are two versions of VRML, 1.0 and 2.0/97. COMSOL Multiphysics only imports VRML 1.0 files.

These two formats do not contain any information about to what face each triangle belongs. Instead, COMSOL Multiphysics uses heuristics to partition the triangles into faces. Usually the partition is reasonable for geometries containing sharp angles between faces, but the partitioning can appear arbitrary for curved geometries such as spheres and tori.

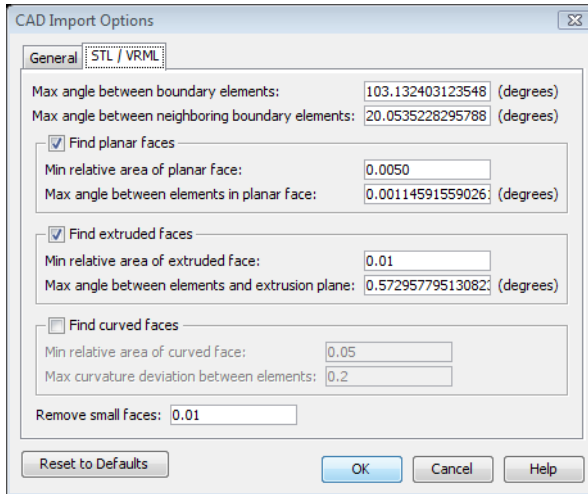
Note: For geometries that you create with an STL or VRML import, the editing capabilities in Draw mode are limited.

Import an STL file by following these steps:

- 1 Open the **Model Navigator** and start a new 3D model.
- 2 Select the menu item **File>Import>Import CAD Data From File**.
- 3 In the **Files of type** list select **STL file**.
- 4 Locate the COMSOL installation directory, go to the **demo** folder, then select the file **block1.STL**.
- 5 Click **Import** to close the dialog box and import the geometry. This action also opens the **Import CAD Data From File** progress window.



In the **Import CAD Data From File** dialog box, click **Options** to open the **CAD Import Options** dialog box. Click the tab **STL / VRML** for import options specific to these surface mesh formats.



You find a detailed description of these options in the section “Importing and Exporting Meshes” on page 364.

Creating a Geometry for Successful Analysis

Several techniques can ensure that a geometry results in a good mesh and gives reasonable solution times for the finite element analysis. They include the use of symmetry, removing unnecessary boundaries, and eliminating gaps, holes, and singularities.

Using Assemblies

When you leave Draw mode COMSOL Multiphysics analyzes the geometry to find all subdomains, boundaries, edges, and points in the geometry. When doing this COMSOL Multiphysics creates an *analyzed geometry* that you apply the physics on. You can choose between two types of analyzed geometries:

- A *composite geometry*. The analyzed geometry object that the software creates is a single object, which effectively is the same object you would get if you select all objects in Draw mode and create the union of them.
- An *assembly*. In this case you get an analyzed geometry that consists of several parts, where each object in Draw mode becomes a part of the assembly.

If there are no problems using a composite geometry this is usually to prefer, but there are some cases when an assembly is the best choice:

- Sometimes the software fails to create a composite geometry. In those cases it might be possible to use an assembly instead. It is a simpler operation to create an assembly, and it often succeeds even if creating a composite geometry fails.
- Each part in an assembly gets a separate mesh. This can make it easier to mesh a complicated geometry, where the mesh generator fails to create a mesh for the composite geometry. Bear in mind that having separate meshes means that the meshes might not be compatible at the two sides of a border between two parts. This can give you a less accurate solution.
- Assemblies make it possible to apply boundary conditions where the dependent variable is discontinuous at the border between two parts. One example is a contact resistance boundary condition for electric currents where the electric potential is discontinuous.

To choose between using a composite geometry and using an assembly, toggle the **Use Assembly** menu item on the **Draw** menu.

For more information about modeling with an assembly, see “Using Assemblies” on page 413 in the *COMSOL Multiphysics Modeling Guide*.

Using Symmetries

Using symmetry is one of the most effective ways to reduce the size of a finite element model. See “Using Symmetries” on page 7 of the *COMSOL Multiphysics Modeling Guide* for advice on modeling with symmetries.

Removing Unnecessary Boundaries

Building a composite solid object from several geometry objects can introduce interior boundaries that split a domain into parts even though the same physical properties prevail in all of them. It is good practice to avoid these unnecessary interior boundaries because the extra points, lines, and surfaces they add cause the mesh generator to create extra mesh elements and might even prevent the mesh generator from producing a valid mesh that respects all boundaries. See “Removing Interior Boundaries” on page 35 for information on how to delete interior boundaries. Sometimes rebuilding a geometry is required to remedy unnecessary interior boundaries.

Making the Geometry Match the Boundary Conditions

In many modeling solutions, the modeling domain is unbounded or too large for successful meshing and analysis. For these cases a suitable boundary condition can replace the exterior of the domain. It is important for the geometry to be large enough to validate these boundary conditions. For outflows in fluid-flow models, for example, the boundary should be perpendicular to the fully developed flow. Inspections and modifications of the solved model might be necessary to verify the validity of the boundary condition.

Avoiding Excessively Small Details, Holes, and Gaps

Many geometries, especially those designed using a CAD system without finite element analysis, contain small holes, details, and gaps. These small features can make the domain unbounded and must be removed before analysis. Small details and holes

can lead to large meshes and finite element models or even failure during mesh generation. To avoid small gaps and mismatches between the geometry objects, make sure the snapping feature is activated.

The DXF import routine has tolerance parameters that fill gaps and holes smaller than the tolerance. See “Importing and Exporting DXF” on page 87 for details.

The CAD Import Module contains tools for automatic and interactive repair and defeaturing of 3D CAD data.

Avoiding Singularities and Degeneracies in the Geometry

A singularity in a geometry is a sharp corner or angle that can create problems during meshing and analysis. In reality, a sharp reentrant corner leads to infinite stress values in a stress analysis of a perfectly elastic material. To avoid a singularity, round sharp corners using fillets. The stress value for a sharp corner is finite in the finite element analysis, but refinement of the mesh increases the stresses in the corner without limit.

A degeneracy in the geometry can occur during solid modeling. For example, fillet areas that taper to a point and the apex of a cone can become degenerate points. These degeneracies might cause problems for the mesh generator and problems during the analysis of the model. A common degeneracy in the geometry occurs when you create a 3D solid (for example, a cylinder) by rotation about an axis that touches the rotation area. In such cases, it is better to create the solid object by extruding a cross section or to use geometric 3D primitives.

Associative Geometry

Associative geometry is a concept for the automatic updating of applied physical properties, such as boundary conditions and equation coefficients, under geometric transformations. Thus, once you have defined the physical properties of a finite element model and return to Draw mode to modify the geometric model, COMSOL Multiphysics updates the physical properties according to the geometry modifications as you exit Draw mode.

The associative geometry functionality utilizes geometry-mapping information between the domain groups (vertices, edges, boundaries, and subdomains) in the *analyzed geometry* and the corresponding domain groups in the geometric model available in Draw mode.

This geometry mapping is not always without ambiguities. Thus, COMSOL Multiphysics makes some heuristic decisions when mapping the physical properties between the analyzed geometric object (the object on which the physical properties are imposed) and the geometric model available in Draw mode. In some cases the resulting updated physical properties might not be the ones that you would expect.

Visualization and Selection Tools

COMSOL Multiphysics provides a number of visualization tools for controlling the view of a model as well as zooming and orbiting around it. Other tools help in selecting domains for the modeling process. Advanced lighting and rendering features can create specialized visualizations. This chapter details how to use these tools.

Visualizing a Model

COMSOL Multiphysics supplies a number of tools for zooming, orbiting, and panning to adjust the view of a model for preprocessing, postprocessing, and general visualization.

When building a model, the right view of a 3D geometry object facilitates not only understanding what the geometry looks like but also selecting features within it. The choice of visualization depends on the object types being rendered and the ones being highlighted. For example, you can view deselected faces as *wireframe plots* or highlight the faces that are selected using rendering options. Furthermore, you control visualization quality through the choice of the geometry mesh detail and lighting options.

For visualization of the results from an analysis, see “Postprocessing and Visualization” on page 453.

- For more information about updating symbols, see “Settings for Load and Constraint Symbols” on page 121.
- For more information about coordinate probe plots, see “Coordinate Probe Plots” on page 484.
- The hide and show objects buttons are shortcuts for the suppression feature described in “Suppressing the Display of Objects” on page 122. Click the **Hide Selected Objects** toolbar button to hide the current selection. Click the **Show All Hidden Objects** toolbar button to make all previously hidden objects of the type corresponding to the current mode visible again. Note that, for instance, edges that you have selected to hide are not made visible again if you are in Subdomain mode and click the **Show All Hidden Objects** button. To make the edges visible again, you must go to Edge mode and click **Show All Hidden Objects**, or use the **Suppress Edges** dialog box available from **Options>Suppress>Suppress Edges**.
- For more information about geometric properties, see “Exploring Geometric Properties” on page 77.

Zooming

The zoom tools function similarly in 1D, 2D, and 3D. All the zoom tools are available from the main toolbar and from the **Options** menu.



Figure 3-1: The zooming toolbar buttons, from left to right: Zoom In, Zoom Out, Zoom Window, Zoom to Selection, and Zoom Extents.

ZOOMING IN AND OUT

To zoom in by a factor of 0.5 or out by a factor of 2, go to the **Options** menu, point to **Zoom**, and then click **Zoom In** or **Zoom Out**. Alternatively, click the corresponding toolbar buttons.

ZOOMING USING A RUBBER-BAND BOX

Manually set the size of the view by zooming with the rubber-band box. To do this, either go to the **Options** menu, point to **Zoom**, and then click **Zoom Window**, or click the corresponding toolbar button. In either case, the mouse defines a zoom window.

ZOOMING TO THE CURRENT SELECTION

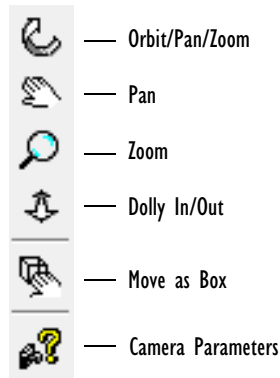
Use the **Zoom to Selection** button to resize your view so that the current selection fills the entire drawing area. This tool is also available from the **Options>Zoom** menu.

ZOOMING TO THE EXTENTS OF THE GEOMETRY

If your geometry is extremely large or small relative to your choice of axis limits, you can quickly resize the view so that the geometry fits into the entire work area. To zoom in or out to the extents of the current geometry, go to the **Options** menu, point to **Zoom**, and then click **Zoom Extents**. Alternatively, use the corresponding toolbar button.

Changing the View in 3D

For 3D models you can click to select an object and then also either orbit, pan, or zoom from your mouse and keyboard. To only select an object but not perform the other operations, click the **Orbit/Pan/Zoom** button in the Camera toolbar once. This also enables selection using a rubber-band box. Click that toolbar button again to enable orbit, pan, and zoom.



To avoid rendering complicated models when orbiting, panning, zooming, or dollying, use the **Move as Box** mode. When **Move as Box** is active, COMSOL Multiphysics represents the geometry using its bounding box while performing the operation. You can specify the default setting of this mode on the **Visualization** page in the **Preferences** dialog box (on the **Options** menu).

ORBIT

Orbit rotates the camera position about the camera target. To orbit camera:

- 1 Click the **Orbit/Pan/Zoom** button to enable orbit.

- 2 Using the left mouse button, click in the drawing area and rotate the camera about the target while holding the mouse button down.

Use the arrow keys to orbit in small increments. Press the **up arrow**, **down arrow**, **left arrow**, or **right arrow** key to rotate the camera in the corresponding direction.

Pan is performed when arrow keys or left mouse button are used while **Ctrl** key is down. **Zoom** is performed when arrow keys or left mouse button are used while **Shift** key is down.

PAN

Pan moves the view horizontally and vertically by moving the camera position and target. To pan camera:

- 1 Click the **Pan** toolbar button to enable pan.
- 2 Using the left mouse button, click and drag the cursor to pan the camera in the corresponding direction.

Use the arrow keys to pan in small increments. Press the **up arrow**, **down arrow**, **left arrow**, or **right arrow** key to pan the camera in the corresponding direction.

ZOOM

Zoom makes objects larger or smaller on the screen by changing the camera view angle. Zooming does not change the camera position. To zoom in and out:

- 1 Click the **Zoom** toolbar button to enable zoom.
- 2 Using the left mouse button, click and move the mouse in the drawing area while holding the mouse button down.

Use the **up arrow** or **down arrow** keys to zoom in small increments.

DOLLY IN AND OUT OF 3D GEOMETRIES



The Dolly In/Out function moves the camera position toward or away from the camera target while keeping the camera view angle constant. This makes it possible to move the camera through objects in a scene. To dolly the camera position in or out:

- 1 Click the **Dolly In/Out** toolbar button to enable this function.
- 2 Using the left mouse button, click and move the mouse in the drawing area while holding the mouse button down.

Alternatively, use the **up arrow** or **down arrow** keys to dolly the camera in or out in small increments.

CONSTRAINED CAMERA MOVEMENT IN 3D

Click the **Camera Parameters** toolbar button to open a dialog box that lets you change how the software constrains camera movement.

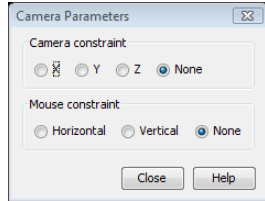


Figure 3-2: The Camera Parameters dialog box.

The **Camera constraint** panel controls the principal axes that the software uses when you move the camera. The principal axes constrain camera motion along lines that are parallel and perpendicular to the principal axis that you select.

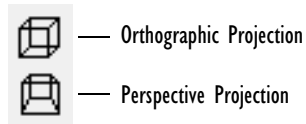
The **Mouse constraint** panel controls mouse movements when you update the view.

Changes in the **Camera Parameters** dialog box take effect immediately.

MOVING BACK AND FORTH IN THE CAMERA HISTORY

Using the **Back in Camera History** and **Forward in Camera History** buttons in the Camera toolbar you can switch between all the camera settings that you previously created using orbit, pan, or zoom commands.

CHANGING THE CAMERA PROJECTION



To change the camera projection, click the **Orthographic Projection** and **Perspective Projection** toolbar buttons.

An orthographic projection (parallel projection) changes the view so that two parallel lines never converge to a single point. Shapes in a drawing remain the same and do not look distorted when they are close to the camera.

A perspective projection changes the view so that all parallel lines converge at one point. Shapes in a drawing appear smaller if they are far from the camera and larger if they are close to the camera. The shapes appear distorted when they are very close to the camera. This projection gives a view that closely resembles what your eye would see.

CAMERA SETTINGS

It is also possible to set all the above camera settings on the **Camera** tab in the **Visualization/Selection Settings** dialog box (in the **Options** menu).

Explicit View Settings

In the **Visualization/Selection Settings** dialog box you can also specify explicit view settings. Click the **Camera** tab and then enter the x , y , and z components for the camera position, camera target, and camera up vector in the corresponding edit fields in the **Explicit View Settings** area. The software normalizes the values in the **Camera up vector** edit fields so that the vector has unit length. You can also specify a view angle in the **Camera view angle** edit field. Click **Apply** or **OK** to change the view according to these settings. Using any of the predefined views in the Visualization/Selection toolbar updates these settings according to the current view.

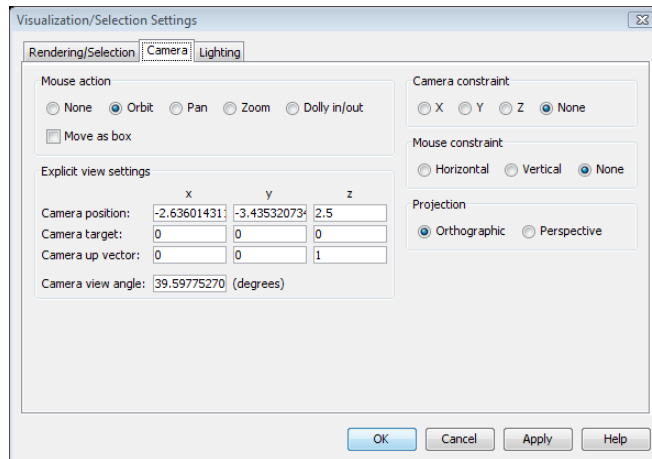


Figure 3-3: The camera settings in the Visualization/Selection Settings dialog box.

Using Lighting

Well-chosen lighting can significantly improve the quality of the visualization for a geometry object, a mesh plot, or a solution plot.

The *headlight* is positioned at the camera and directed toward the target. The *scene light*, on the other hand, radiates from a far distance to simulate light from the sun.

Lighting is available only in the 3D view (3D models as well as height plots during postprocessing of 2D models). Use the Camera toolbar to switch light sources on or off:

- Click the **Scene Light** button to turn the scene lights on or off.
- Click the **Headlight** button to control a light source positioned at the camera. When the camera position changes, the headlight follows.

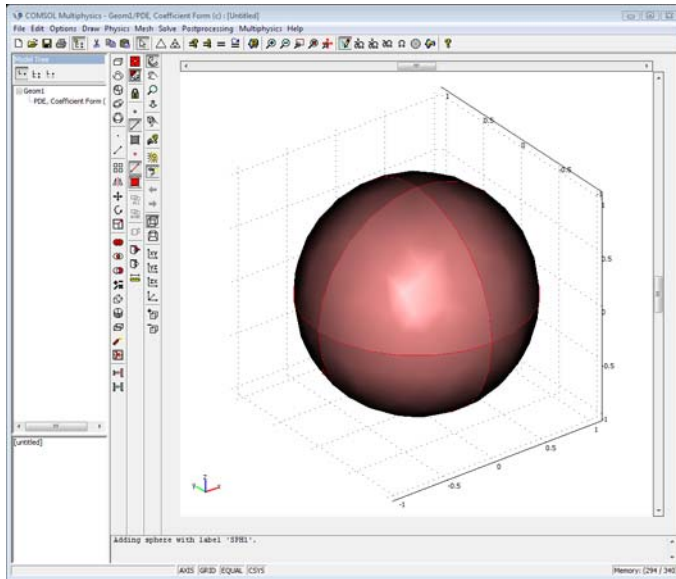


Figure 3-4: Using a headlight on a 3D geometry.

You can also turn light sources on and off and control light properties using the **Visualization/Selection Settings** dialog box. From the **Options** menu, choose **Visualization/Selection Settings**; in the dialog box, click the **Lighting** tab.

HEADLIGHT SETTINGS

Turn on the headlight by selecting the **Headlight** check box.

To change the headlight's color, click the **Color** button. Make a selection in the **Light Color** dialog box and click **OK**.

SCENE LIGHT SETTINGS

Four scene lights are installed by default. You can turn each of them on or off by selecting or clearing the **Enabled** check box. You can also specify their color by clicking the **Color** button and making a selection in the **Light Color** dialog box.

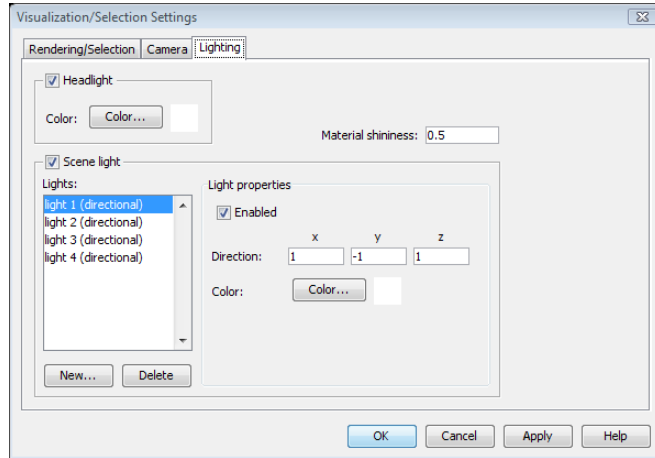


Figure 3-5: The default lighting settings in the Visualization/Selection Settings dialog box.

Adding a Scene Light

- 1 Open the **Visualization/Selection Settings** dialog box and click **New**.
- 2 In the **New Light** dialog box select the type of light source from the **Type** list: directional light, point light, spotlight, or ambient light.
- 3 Click **OK**.

Removing a Scene Light

Select a scene light and click **Delete**.

Settings for Directional Lights

The predefined light sources are all directional lights. To create a new directional light, select **Directional** in the **Type** list. Doing so creates one with its origin at infinity. Specify the light's direction vector using the **x**, **y**, and **z** edit fields.

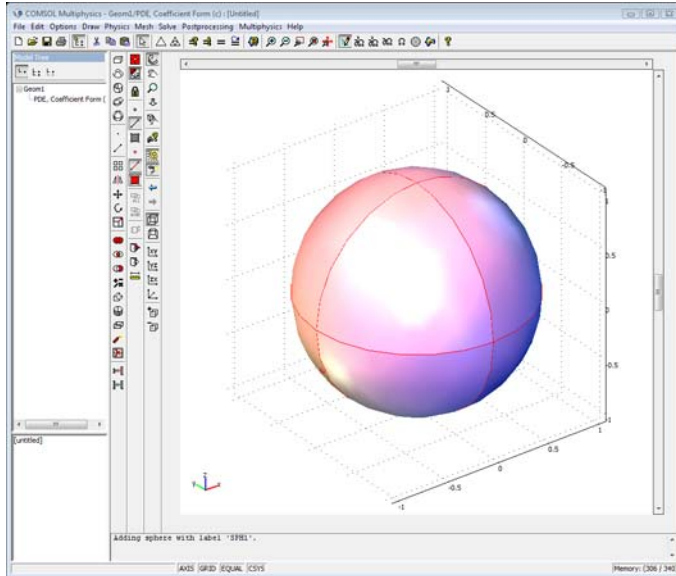


Figure 3-6: Directional scene light using yellow and blue lights.

Settings for Point Lights

To create a point light, go to the **New Light** dialog box and select **Point** in the **Type** list. A point light is a source at a fixed point in space that radiates light equally in all directions away from itself. Specify the light's position vector using the **x**, **y**, and **z** edit fields.

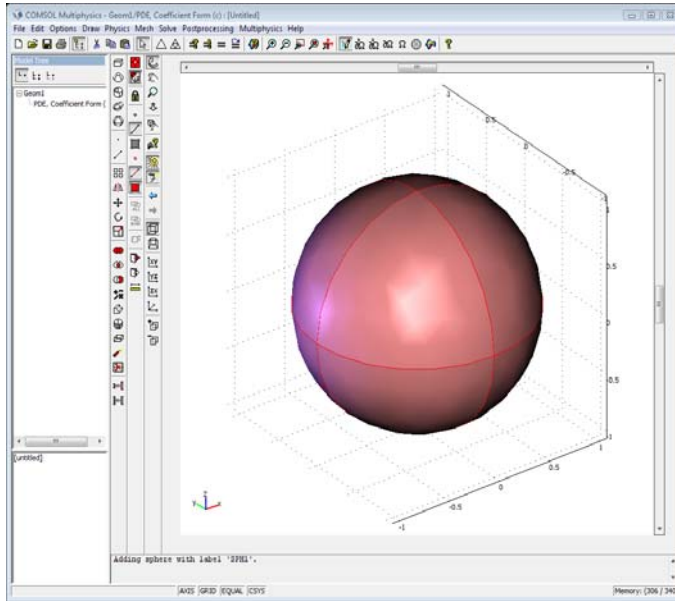


Figure 3-7: A point light is positioned to the left of the sphere.

Settings for Spotlights

To create a spotlight, go to the **New Light** dialog box and select **Spot** in the **Type** list. A spotlight is a source at a fixed point in space that radiates light in a specified direction.

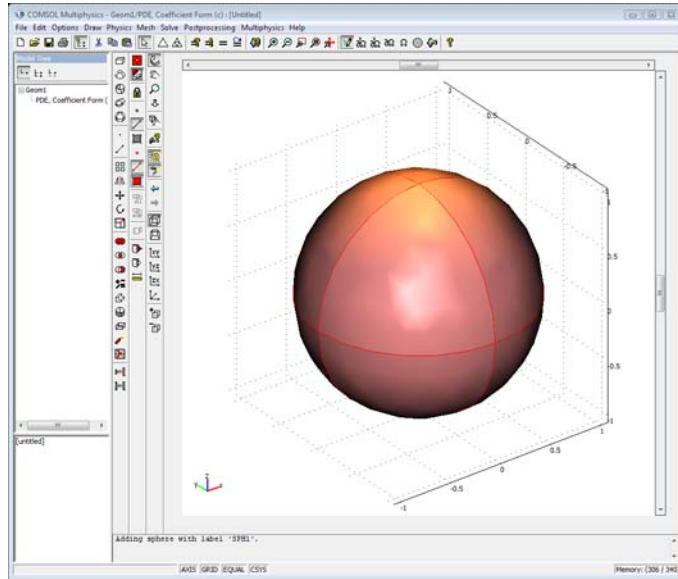


Figure 3-8: A yellow spotlight positioned above the sphere.

To define a spotlight:

- 1 Open the **Visualization/Selection Settings** dialog box.
- 2 Click the **Lighting** tab and select the spotlight you wish to define.
- 3 Specify the position using the **Position** edit fields (under **x**, **y**, and **z**).
- 4 Specify the light's direction vector using the **Direction** edit fields (under **x**, **y**, and **z**).
- 5 Specify the spread angle in degrees using the **Spread angle** edit field. This is the angle between the direction axis and a ray along the edge of the cone of light coming from the source. The allowed spread angle can vary between 0 and 90 degrees; the default is 90 degrees.
- 6 Specify the concentration of the spotlight as a value between 0 and 1 in the **Concentration** edit field. This parameter specifies how quickly the light intensity attenuates as a function of the angle of radiation as measured from the direction of radiation. A value of 1 creates the most focused light beam.

7 Specify the color as done with the other light sources.

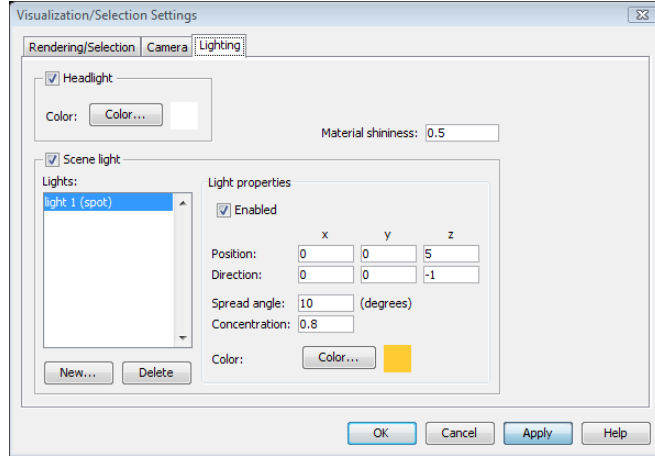


Figure 3-9: The lighting settings for a spotlight.

Settings for Ambient Light

To create an ambient light, select **Ambient** in the **Type** list. An ambient light seems to come from all directions. In this case color is the only property.

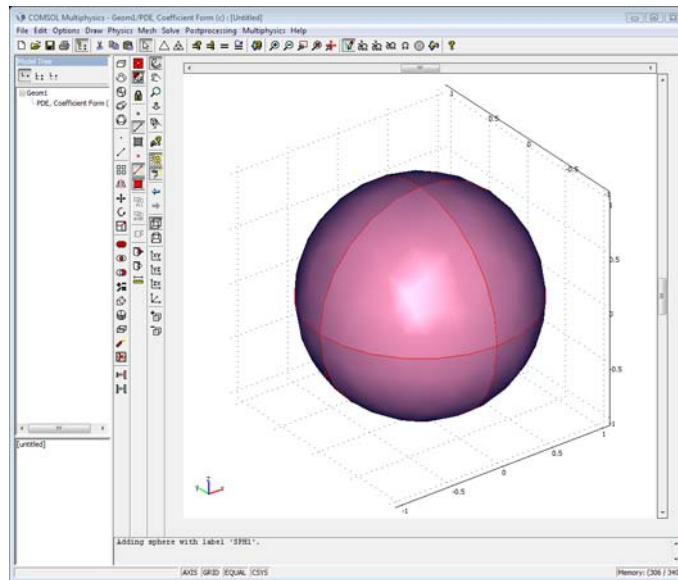


Figure 3-10: Lighting of the sphere using an ambient light.

SPECIFYING MATERIAL SHININESS

Enter a value between 0 and 1 in the **Material Shininess** edit field; 0 means not shiny (dull), and 1 is very shiny.

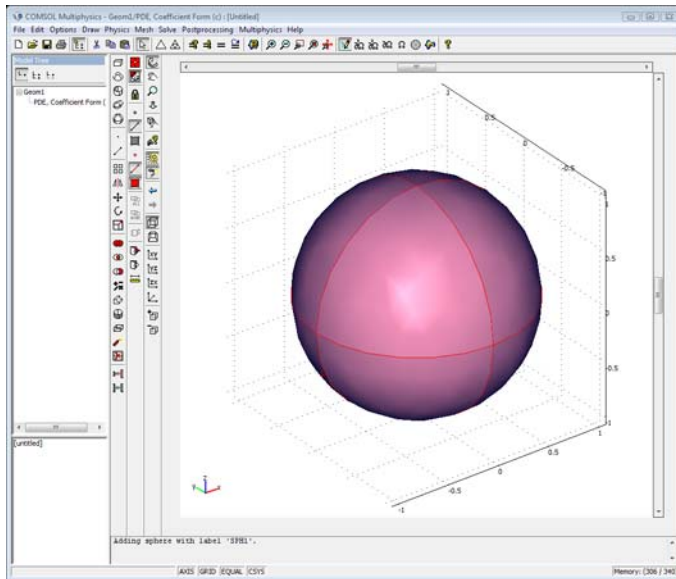
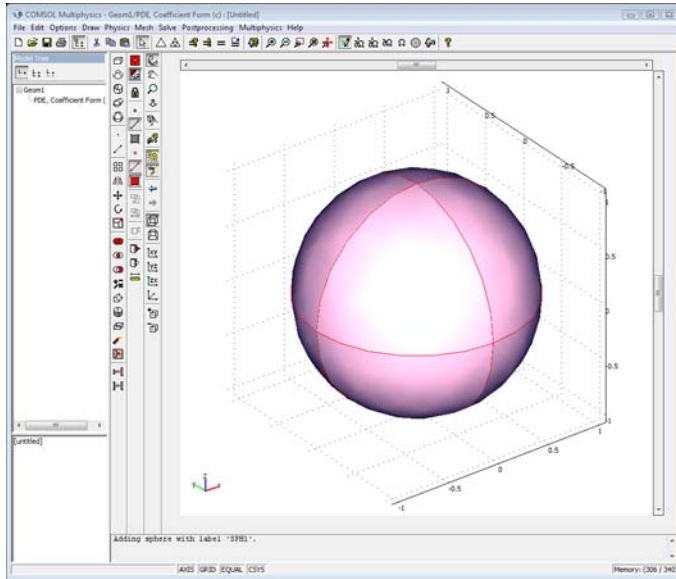


Figure 3-11: Very shiny material (above) and dull material (below).

If you think a plot is too bright or dark, try to change the shininess, disable some lights, or use darker or brighter light colors.

Changing the View to a Different Axis Plane

Four buttons in the Camera toolbar provide quick settings to change the view to the various axis planes:

- Click the **Go to XY View** button to change the view to the xy -plane.
- Click the **Go to YZ View** button to change the view to the yz -plane.
- Click the **Go to ZX View** button to change the view to the zx -plane.

The first click on one of these buttons selects the plane view with a positive normal direction. A second click on the same button switches to a negative normal direction.

- Click the **Go to Default 3D View** button to change the view to the default.

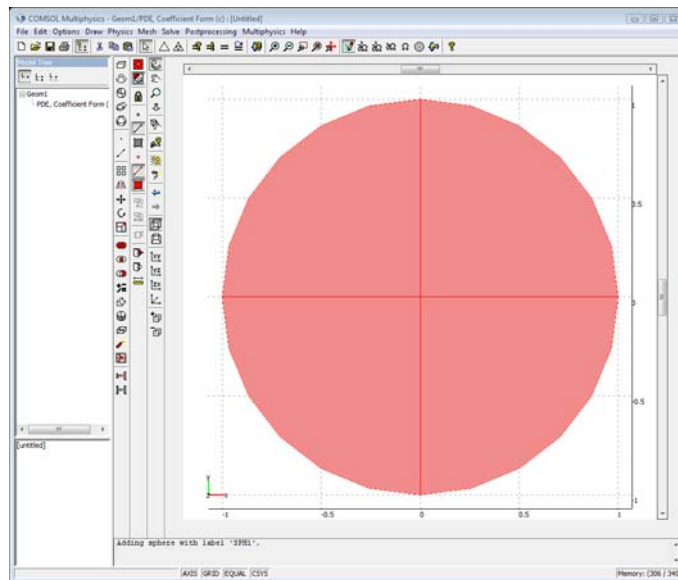


Figure 3-12: A zx -plane view of a sphere.

Using Transparency in 3D Plots

Use the bottom two buttons in the Camera toolbar to change the geometry objects' transparency. These buttons modify the *alpha-blending value*, which varies from 1

(no transparency—fully opaque) to 0 (maximum transparency—invisible). Change the settings as follows:

- Click the **Increase Transparency** button to decrease the alpha-blending value in steps of 0.1.
- Click the **Decrease Transparency** button to increase the alpha-blending value in steps of 0.1.

You can also set a value for transparency in the **Visualization/Selection Settings** dialog box:

- 1 From the **Options** menu choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 Enter an alpha-blending value in the **Transparency** edit field.
- 4 Click **OK**.

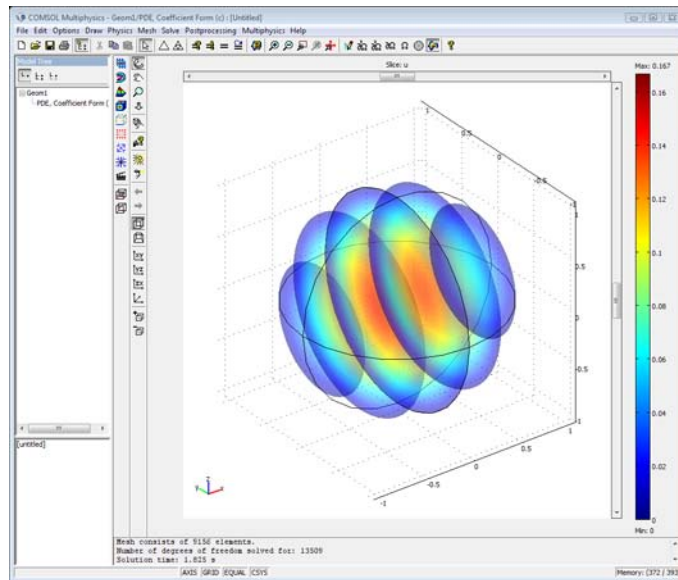


Figure 3-13: Slice plot using transparency for a see-through effect.

Controlling the Resolution of the Visualization Mesh

The resolution of the visualization mesh affects the visualization of geometry objects during modeling and preprocessing in several modes: Edge mode, Boundary mode,

and Subdomain mode. The resolution setting controls the number of polygons that COMSOL Multiphysics uses when rendering geometry surfaces.

You can select a finer mesh to better resolve a geometry's curvature or select a coarser mesh to reduce memory requirements for rendering. To change the visualization mesh resolution:

- 1 From the **Options** menu, choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 Select **Fine**, **Normal**, or **Coarse** in the **Visualization mesh** list.
- 4 Click **OK**.

Note: The visualization mesh is independent of the mesh used for the finite element analysis. Changes to the visualization mesh do not affect analysis results.

Adding Labels to Domains

You can add labels to the following domain types:

- Geometry objects—these labels are visible only in Draw mode.
- Vertices—these labels are visible only in Point mode.
- Edges (3D only)—these labels are visible only in Edge mode.
- Boundaries—these labels are visible only in Boundary mode.
- Subdomains—these labels are visible only in Subdomain mode.

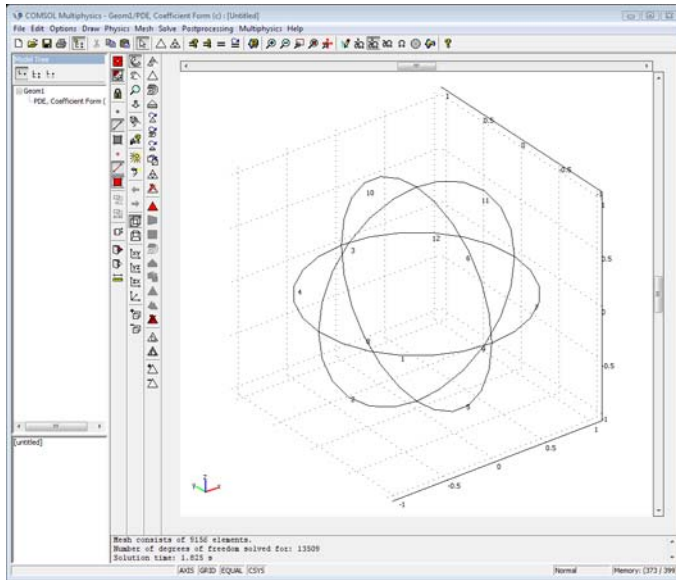


Figure 3-14: Edge labels.

To turn labels on and off, go to the **Options** menu, point to **Labels**, and then place a check mark next to **Show Geometry Object Labels**, **Show Vertex Labels**, **Show Edge Labels**, **Show Face Labels**, or **Show Subdomain Labels**.

To turn labels on and off using the **Visualization/Selection Settings** dialog box:

- 1 From the **Options** menu choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 Under **Labels** select the check boxes for the labels you want to use.
- 4 Click **OK**.

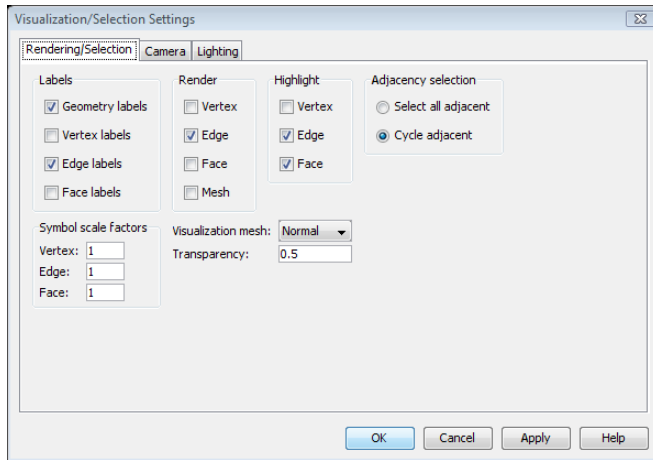


Figure 3-15: Rendering and selection settings in the Visualization/Selection Settings dialog box.

Rendering and Highlighting Geometry Objects

COMSOL Multiphysics creates a geometry object using *subdomains*, *faces*, *edges*, and *vertices*. For example, a cube consists of one subdomain surrounded by six faces. The faces are bound by twelve edges that connect at eight vertices. Thus it is possible to visualize a cube by displaying one or more of these four types. For instance, you can generate a wireframe plot by rendering only the cube’s edges.

Geometry objects are *adjacent* if they connect directly to each other. Hence all faces, edges, and vertices on the cylinder are adjacent to the subdomain. An edge on the cylinder is adjacent to two faces and two vertices.

COMSOL Multiphysics allows control of the rendering and highlighting of geometry objects. Only geometry objects that are visible (rendered) are clickable so that you can use them for selection purposes. COMSOL Multiphysics highlights selected geometry objects; in the default settings, it renders edges and then highlights edges and faces. The rendering and highlighting settings are independent of each other.

CONTROLLING THE RENDERING OF GEOMETRY OBJECTS

To enable or disable the rendering of vertices, edges, and faces, either use the buttons in the Visualization/Selection toolbar or work with the **Visualization/Selection Settings** dialog box.

Controlling Rendering Using the Visualization/Selection Toolbar

- The **Render Vertex** button activates or deactivates the rendering of vertices.
- The **Render Edge** button activates or deactivates the rendering of edges.
- The **Render Face** button activates or deactivates the rendering of faces.

Controlling Rendering Using the Visualization/Selection Settings Dialog Box

- 1 From the **Options** menu choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 Under **Render**, select or clear the **Vertex**, **Edge**, and **Face** check boxes to set the rendering for these geometry objects.
- 4 Click **OK**.

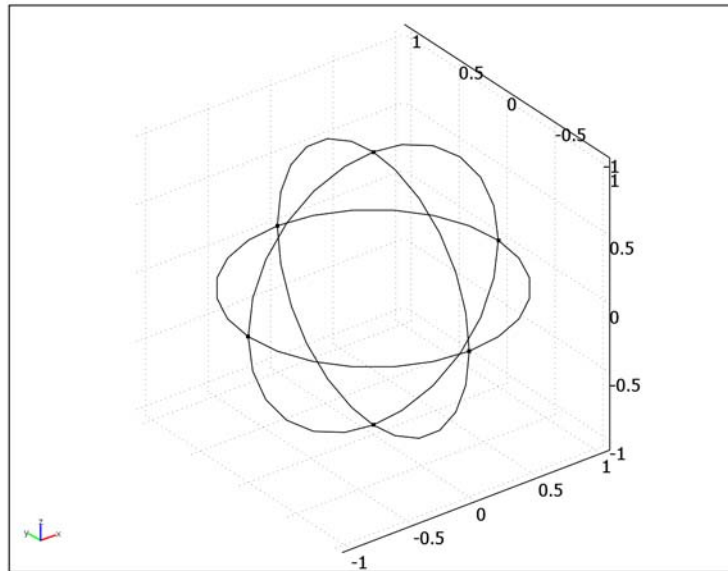


Figure 3-16: Rendering of vertices and edges.

Enabling and Disabling Mesh Rendering

Under **Render**, select or clear the **Mesh** check box to turn mesh visualization on and off in the domain selection modes. This check box corresponds to the **Interactive Meshing** button on the Mesh toolbar.

CONTROLLING THE HIGHLIGHTING OF GEOMETRY OBJECTS

To enable or disable the highlighting of vertices, edges, and faces, either use the buttons in the Visualization/Selection toolbar or work with the **Visualization/Selection Settings** dialog box.

Controlling Highlighting Using the Visualization/Selection Toolbar

- The **Highlight Vertex** button activates or deactivates the highlighting of vertices.
- The **Highlight Edge** button activates or deactivates the highlighting of edges.
- The **Highlight Face** button activates or deactivates the highlighting of faces.

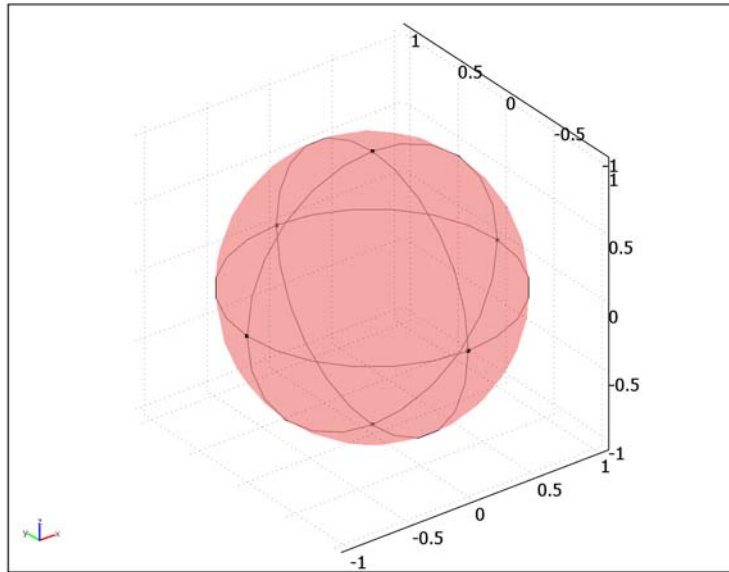


Figure 3-17: Highlighting of faces only. Transparency settings show the rendering of vertices and edges.

Controlling Highlighting Using the Visualization/Selection Settings Dialog Box

- 1 From the **Options** menu choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 Under **Highlight** select or clear the **Vertex**, **Edge**, and **Face** check boxes to set the highlighting for these geometry objects.

4 Click **OK**.

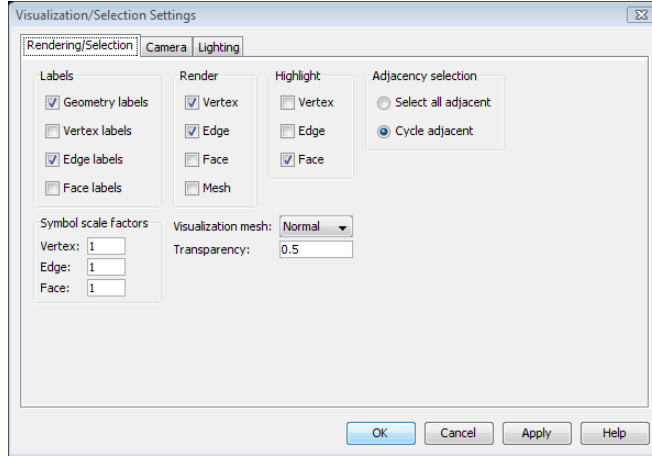


Figure 3-18: Dialog box settings for the rendering of vertices and edges and the highlighting of faces.

Scaling of Load and Constraint Symbols

In the **Symbol scale factors** area you can set scale factors for the symbols that indicate loads and constraints when using the Structural Mechanics Module and the MEMS Module. Individual scale factors are available for symbols on vertices, edges, and faces.

Saving Preferences for Labels, Rendering, and Highlighting

Use the **Preferences** dialog box to set preferences for labels, rendering, and highlighting. The settings in this dialog box are active immediately, but COMSOL Multiphysics also saves them as preferences for later modeling sessions.

- 1 From the **Options** menu choose **Preferences**.
- 2 In the **Preferences** dialog box click the **Visualization** tab.
- 3 In the **Show** area select the check boxes for the label types and other graphical objects to display.
- 4 In the **Render** and **Highlight** areas select the check boxes for the geometry objects you want rendered and highlighted. Geometry objects are highlighted when selected.

5 Click **OK**.

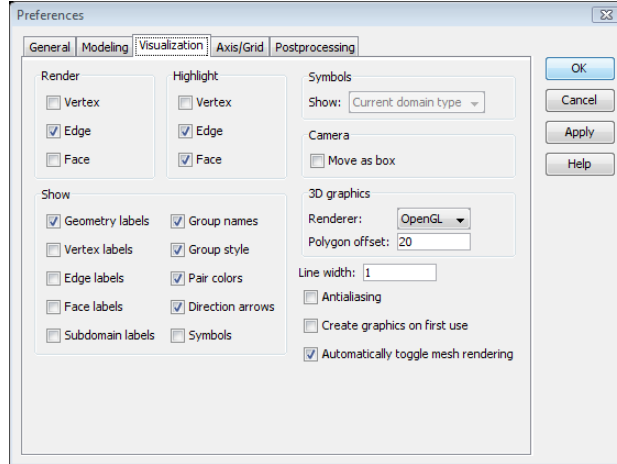


Figure 3-19: The settings for visualization properties in the Preferences dialog box.

USING THE POLYGON OFFSET FOR IMPROVED 3D RENDERING

For certain graphics cards and drivers, a rendering problem might appear when combining faces and edges. The symptom is unsightly line-rendering lines that seem almost dotted. To fix this 3D rendering problem use the preferences setting for polygon offset. Try the following if you believe 3D geometry objects or the mesh are suffering from this problem:

- 1 Start a 3D model and add a cone.
- 2 Go to the **Preferences** dialog box and click the **Visualization** tab.
- 3 In the **3D graphics** area, enter a suitable value in the **Polygon offset** edit field. The default value is 2. Sometimes you must increase it to 10, 1000, or an even higher value. Be aware, though, that an excessively high value for the polygon offset results in strange plots where edges seem to be drawn on top of faces.
- 4 Click **OK**.

LINE RENDERING SETTINGS

You can control how lines appear on the screen using the **Line width** and the **Antialiasing** settings on the **Visualization** tab in the **Preferences** dialog box. You can specify the line width in pixels and if antialiasing should be used or not. Antialiasing is a software technique for diminishing stairstep-like lines to make the lines look smoother.

Note: Rendering can be slow on some graphics cards when antialiasing is used. Some graphics cards do not support antialiasing, which means that the **Antialiasing** setting has no effect on rendering.

SELECTING WHEN TO CREATE GRAPHICS

By selecting or clearing the **Create graphics on first use** check box you can choose if graphics data should be created when it is first needed or if it should be created when the model is initially loaded.

AUTOMATICALLY TOGGLE MESH RENDERING

If you select the **Automatically toggle mesh rendering** check box the software automatically enables the **Interactive Meshing** button on the Mesh toolbar when you do an interactive meshing operation and disables the button when you open any physics settings dialog box. Clear the check box to disable this behavior. For more information on the **Interactive Meshing** toolbar, see “Using the Mesh Toolbar” on page 346.

SETTINGS FOR LOAD AND CONSTRAINT SYMBOLS

Note: The symbol settings and display of symbols are only supported by the application modes for structural mechanics in the Structural Mechanics Module, the MEMS Module, and the Acoustics Module. See the documentation for those modules for specific information about symbols for loads and constraints.

When you use COMSOL Multiphysics together with the Structural Mechanics Module, the MEMS Module, or the Acoustics Module, there is a **Symbols** area on the **Visualization** tab in the **Preferences** dialog box with settings controlling the behavior of the load and constraint symbols. The **Show** list is enabled when the **Symbols** check box in the **Show** area is selected. In the **Show** list, select:

- **Current domain type** to show only the symbols for the current domain type. For example, if you are in Boundary mode, this setting only renders symbols for loads and constraints set on boundaries.
- **All** to show symbols for all domain types, independent of what mode the user interface is in.

Note: The polygon offset setting affects 3D rendering only when your PC is using OpenGL; this technique is not useful if you are using DirectX.

Suppressing the Display of Objects

If you want to select objects buried deep inside a complex geometry, it can be useful to suppress parts of a geometry. The suppression feature makes selected objects invisible according to the visualization settings. By suppressing objects you can reveal hidden objects or clear the way to access objects that otherwise would be difficult to reach.

You can suppress the following types of geometry objects and domains:

- Geometry objects (for geometry modeling in Draw mode)
- Subdomains
- Boundaries
- Edges

To suppress geometry objects:

- 1 Go to the **Options** menu, point to **Suppress**, and place a check mark next to **Suppress Geometry Objects**.
- 2 In the **Suppress Geometry Objects** dialog box, select the objects you want to suppress from **Object selection** list.
- 3 Click **Apply**.

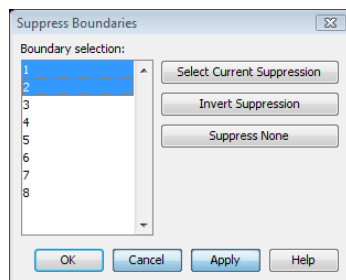


Figure 3-20: The Suppress Boundaries dialog box for suppression of boundaries.

To select all suppressed geometry objects, click the **Select Current Suppression** button.

Clicking the **Invert Suppression** button suppresses objects that are not currently suppressed and vice versa.

To remove all suppressions, click the **Suppress None** button or clear the selection from all objects in the list.

Geometry object suppression affects only Draw mode.

Work in the same fashion with the **Suppress Subdomains**, **Suppress Boundaries**, and **Suppress Edges** dialog boxes.

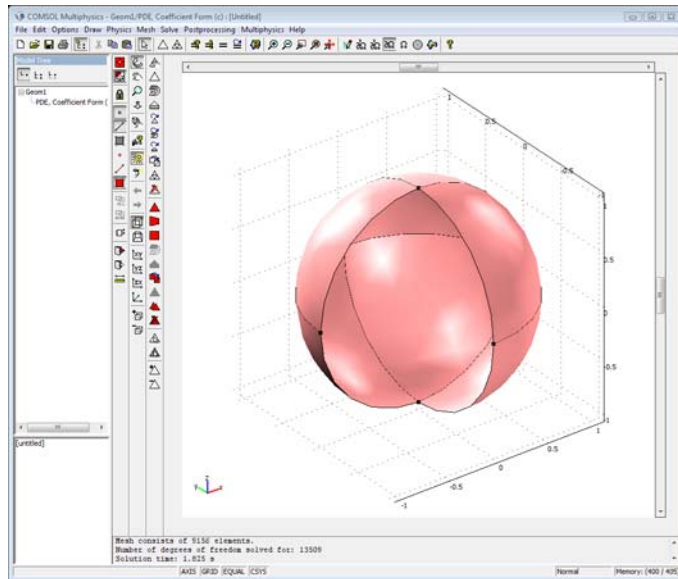


Figure 3-21: The display of the sphere faces after suppressing boundaries 1 and 2.

Rendering Large Geometry Objects

This section describes some memory-saving techniques that can be useful when working with large models on a computer with limited processor capacity and memory.

RENDERING A WIREFRAME PLOT ONLY

One way to conserve memory is to render as little as possible of the geometry. Face objects are the most memory-consuming objects. To remove all face rendering, turn off both rendering and highlighting of faces. To get a *wireframe plot* of the geometry make sure the edges are rendered.

USE A COARSE VISUALIZATION MESH

Using a coarse visualization mesh saves memory for rendering. This affects only the visualization of the geometry objects. For information about setting the visualization mesh detail see “Controlling the Resolution of the Visualization Mesh” on page 113.

RENDERING AND THE JAVA HEAP SPACE

Java allocates memory on a heap. This heap is initially quite small and then grows as more memory is needed. When a Java program such as COMSOL Multiphysics is launched, the maximum size for the Java heap must be specified. In COMSOL this value, called MAXHEAP, is set to 256 MB by default. If Java grows the size of the heap past MAXHEAP, you receive an out-of-memory error message, something that can happen when postprocessing or visualizing the mesh for large models. To change the value of MAXHEAP, do as follows:

- On Windows, change the MAXHEAP variable in the file `comsol.opts`, which is located in the `bin` directory under your COMSOL directory.
- On UNIX, Linux, and the Macintosh, edit the `comsol` shell script, located in the `bin` directory under your COMSOL directory.

Note: The memory on the Java heap does not include memory usage in assembly routines and solvers.

Viewing Multiple Geometries

When working with several geometries, you can display the active geometry simultaneously with the outlines of several other geometries. To make other geometries visible:

- 1 Open the **View Geometries** dialog box from the **Options** menu.
- 2 In the **Visible geometries** list select those geometries that should be visible together with the active geometry.
- 3 Click **OK**.

You can also view all geometries or hide all other geometries by clicking the **View All 3D Geometries** or the **Hide Other 3D Geometries** buttons in the Visualization/Selection toolbar (and the corresponding buttons for 2D geometries).

The outlines of the other geometries are visible during all modeling stages. To remove them, open the **View Geometries** dialog box again and clear the selections by Ctrl-clicking in the **Visible geometries** list. Click **Apply** or **OK** to remove the outlines.

Selecting Domains

COMSOL Multiphysics offers several ways of selecting domains when assigning values and expressions to define a model:

- Select by clicking or enclosing the domains in the drawing area
- Select from a list in a dialog box
- Select using a shortcut key

Specific selection methods vary depending on if the model is in 1D, 2D, or 3D.

General Object Selection Methods

The following principles apply for selecting any object:

- To select a single geometry object, click it using the left mouse button.
- To select several objects or deselect some of them, Shift-click the objects.
- To open an object's dialog box, double-click the object with the left (2D) or right (3D) mouse button.
- Clicking outside of the objects deselects all of them.
- To select all objects, choose **Select All** from the **Edit** menu or press Ctrl+A.
- To clear all selections, choose **Deselect All** from the **Edit** menu or press Ctrl+D.

You can also use a rubber-band box to select objects:

- The selection applies only to objects that the rubber-band box surrounds completely.
- Shift-clicking adds the rubber-band box selection to the current selection.
- In 2D, Ctrl-clicking performs an exclusive-OR operation with the current selection (presently selected objects becomes deselected when inside the rubber-band box and vice versa).

In 3D, Ctrl-clicking performs a logical AND operation with the current selection so that only the presently selected objects inside the rubber-band box stay selected.

This makes it possible to select a detail from several selected objects.

To enable the selection function in 3D using a rubber-band box, deselect the **Orbit/Pan/Zoom** button.

SELECTION INDICATION

COMSOL Multiphysics indicates selected objects with highlighting (see “Rendering and Highlighting Geometry Objects” on page 116). Additionally, selected lines have a thicker line width, and selected points are larger. The selection indications occur whether highlighting is active or not.

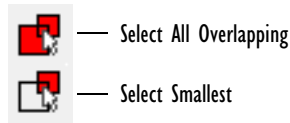
SELECTING IN A LIST

In many dialog boxes you can select objects from a list. A direct correspondence exists between selections in a list and selections in the drawing area. Further, any list selection overrides any confirmed selections in a 3D model. Pressing Ctrl+A (select all selections) or Ctrl+D (clear all selections) also work in lists.

SELECTING BY GROUP

Set the **Select by group** check box in the physics settings dialog boxes to make the selection of one domain also select all other domains with identical settings. This is useful when working with many domains but using only a few different equation settings or materials.

Object Selection Methods in 2D



In a 2D geometry model, choose the selection method using the following geometry selection toolbar buttons:

- **Select All Overlapping:** With this feature active, clicking at the intersection of several objects selects all of them
- **Select Smallest:** With this feature active, clicking at the intersection of several objects selects the smallest of them

You can also click the boundary of a geometry object to select it alone. This is useful, for example, to select a small object embedded in a larger object.

Object Selection Methods in 3D

For 3D objects, similar principles apply for selecting geometry objects, subdomains, faces, edges, and points.

SELECTING A SINGLE OBJECT

Use the left mouse button to select only one object. COMSOL Multiphysics highlights selected objects in red. Other selection methods can be more efficient in certain circumstances. For example, with a face object you can select it by:

- Clicking the face object or its label
- Clicking the face object's vertices or edges (see “Adjacency Selection Methods” on page 134 for details)
- Clicking the labels of the adjacent subdomains

To click on an object you must have already set its object type to be clickable. The rendering and highlighting settings also impose certain restrictions (see “Rendering and Highlighting Geometry Objects” on page 116).

OVERLAPPING OBJECTS

When you click on a 3D geometry, several objects might overlap at the cursor. What the click is meant to choose is then ambiguous. How does COMSOL Multiphysics handle this situation?

An object's priority depends on its type. Labels have the highest priority, then vertices, edges, and finally face objects. As a result, you can always select a point or a curve object even if a face object in the foreground covers it.

Ordering within an object type exists only for faces. Faces are initially sorted based on their area:

- The smallest face has highest priority, so the first click on overlapping faces selects the smallest one.
- A subsequent click selects the second smallest face under the cursor, and so on.

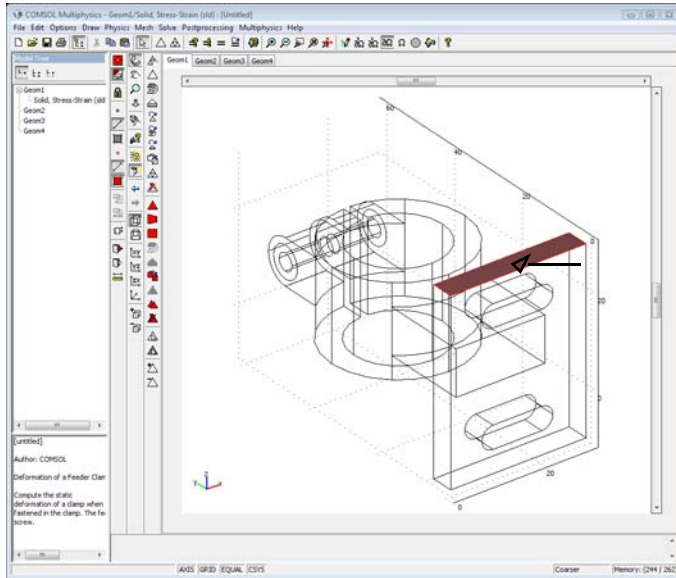
Multiple clicking on overlapping faces cycles among them.

Selecting a Single Face Element

This example demonstrates the selection of a single face segment and object precedence when clicking on overlapping faces.

- 1 In the **Model Navigator** click the **Model Library** tab.
- 2 Open the **COMSOL Multiphysics** folder, then the **Structural Mechanics** folder, and select the **feeder clamp** model.
- 3 Click **OK**.
- 4 Click the **Boundary Mode** button in the Main toolbar.

- 5 Click the **Go to Default 3D View** button.
- 6 Click the **Headlight** button to turn off the lighting.
- 7 Click the **Pan/Zoom/Rotate** button to disable mouse movements during this example.
- 8 Move the mouse cursor to the position indicated by the arrow in the following figure and click once with the left button. As illustrated, two faces overlap at the cursor position, and your click selected the smaller of them.



The face and its adjacent edges are highlighted in red. Click again using the left mouse button without moving the mouse; doing so selects the larger face behind the previously selected face. If you continue to click the same spot, the selection cycles between the two faces.

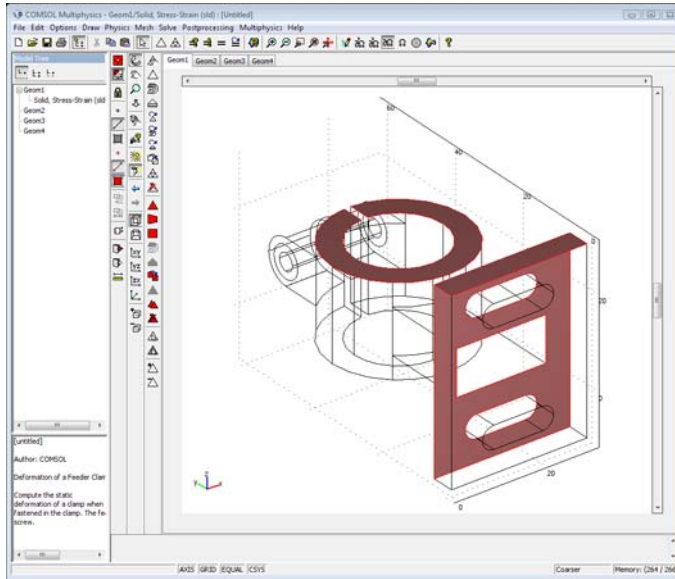
SELECTING MULTIPLE OBJECTS

By Shift-clicking during selection you can include another object in the current selection. Making multiple selections by Shift-clicking is not always satisfactory when several objects are overlapping. To avoid such difficulties you can confirm selections as described in the next section.

Selecting Multiple Face Element

Continuing from the previous example, hold down the Shift key and click two times with the left mouse button in the same location as before; both overlapping faces are

now selected. Move the mouse to other faces and Shift-click on them to add them to the current selection. The following figure shows a selection of four faces:



CONFIRMING SELECTIONS

The confirmation feature is a way to lock in the current selection so that it is active until you explicitly clear the selection. Previously selected objects (highlighted in red according to the visualization settings) turn blue when confirmed. To confirm and lock in a selection:

- Click anywhere in the drawing area, either by using the right mouse button or by holding the **Ctrl** key when clicking; or
- Click the **Confirm Selection** button in the Visualization/Selection toolbar.

Locking in a selection prevents it from being removed by accident when you are making multiple selections among many overlapping objects.

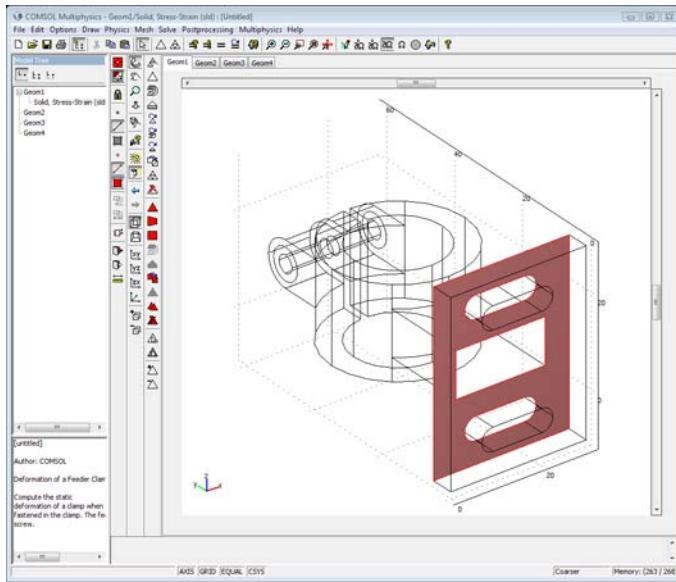
You can reselect confirmed selections in order to mark them for removal. They are then highlighted in green. When you confirm a green selection, the objects become deselected. Red and green selections can exist at the same time as blue selections. A confirm-click at this point removes green selections, and red selections turn blue.

A selection's color—red, green, or blue—does not matter from a modeling point of view. A selected object is always selected, whether it is confirmed or not.

Confirming a Selected Face

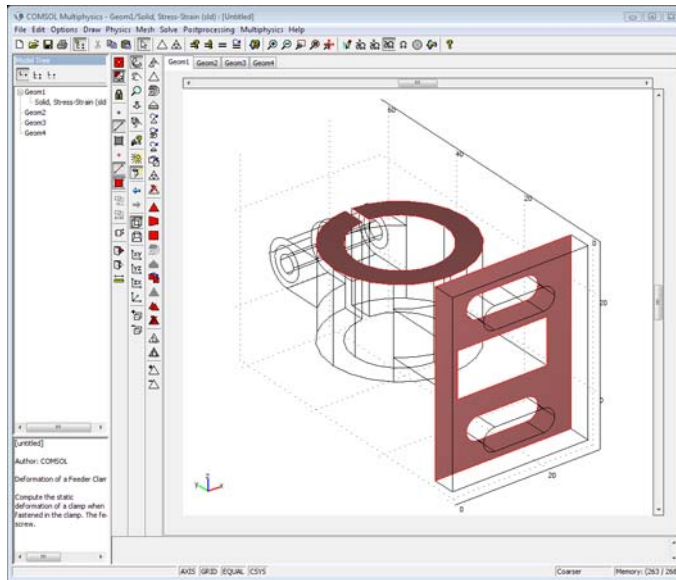
Proceeding from the previous example:

- 1 Click the same spot as in the single-selection example until you have selected the larger of the two faces.
- 2 Confirm the current single selection by clicking with the right mouse button anywhere in the drawing area. The selected face now turns blue.



The selection of this face segment is now locked, and it will not become deselected when you make additional single or multiple selections.

- 3 Click on the top face of the middle cylindrical part of the clamp using the left mouse button and also confirm that selection by right-clicking. The selection now looks like this:



CLEARING SELECTIONS

Clicking outside a geometry with the left mouse button clears (deselects) all unconfirmed selections whereas confirmed selections remain intact. Clearing the selection of a subset of confirmed (blue) selections is possible by selecting the desired objects with the left mouse button (the blue objects turn green) and removing them with a Ctrl-click or by clicking the right mouse button.

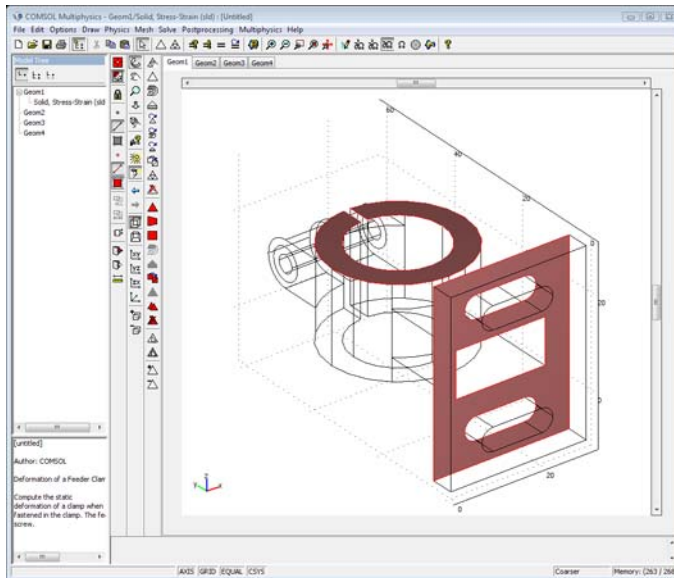
Clearing Selected Faces

Continuing from the previous example:

- 1 Click on the bottom face segment of the cylindrical subdomain in the middle of the clamp. Two blue faces and one red face are now selected.
- 2 Click outside the geometry using the left mouse button and the red selection disappears. However, the confirmed selections remain.

- 3 To remove the confirmed selection on the top of the cylindrical subdomain, click the blue circular-shaped face until it turns green. You might need to click twice if the first click results in a selection of an underlying face object.

The geometry selections now have the following appearance:



- 4 Click in the main drawing area using the right mouse button to remove the green selection. Only one selection remains. Shift-click anywhere on the geometry to add more selections. Press Ctrl+D to deselect all faces.

CLICK-AND-DRAG IN 3D

You can make selections and rotate a geometry without clicking any buttons to change between the two operations. It can be convenient to work with both selections and rotations at the same time. In fact, any mouse action can be active at the same time as selection. Selection is always available, but you can deactivate mouse actions by clicking the **Orbit/Pan/Zoom** button again to deactivate the mouse-movement action. This also enables rubber-band box selection.

In addition, the combination of the two makes it possible to cancel a pending selection. The object does not become selected until you release the mouse button. If you move the mouse before releasing the button, the object is never selected; instead, the mouse-movement action is carried out.

ADJACENCY SELECTION METHODS

You can select 3D objects by clicking them or their adjacent objects. However, an adjacent object is often next to more than one object. In such a case, you can either choose to select all adjacent objects or cycle through them (multiple clicking selects adjacent objects one at a time). These two methods of selection are called adjacency selection methods; selecting by cycling through adjacent objects is the default.

Choosing the Adjacency Selection Method

Choose between the two methods using the **Cycle Adjacent** button and the **Select All Adjacent** button on the top of the Visualization/Selection toolbar (see the example below), or use the **Visualization/Selection Settings** dialog box:

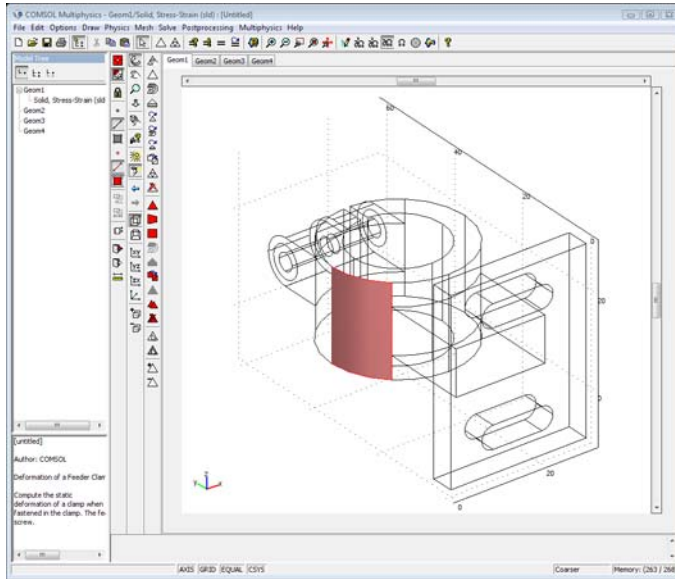
- 1 From the **Options** menu choose **Visualization/Selection Settings**.
- 2 In the **Visualization/Selection Settings** dialog box click the **Rendering/Selection** tab.
- 3 In the **Adjacency selection** area click either the **Select all adjacent** or the **Cycle adjacent** button.
- 4 Click **OK**.

Selecting Using Adjacent Objects

Continuing from the previous example:

- 1 Display the boundaries and turn on vertex rendering by selecting the **Render Vertex** button in the Visualization/Selection toolbar.
- 2 At the top of the Visualization/Selection toolbar make sure that the **Cycle Adjacent** button is selected in the first group of buttons.
- 3 In the **Options** menu, point to **Labels**, and click **Show Vertex Labels**.

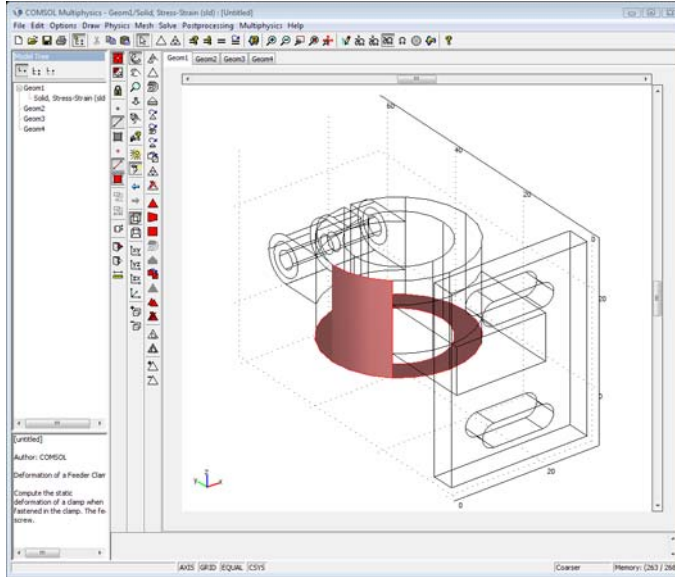
- 4 With the left mouse button click once on Vertex 5 (on the lower left of the middle cylindrical object) and you should see the following:



One of the face segments adjacent to the selected vertex is now selected.

- 5 Click Vertex 5 again. This action selects another of the adjacent faces.
- 6 A third click on the same vertex selects the last of the three adjacent faces, and repeated clicking continues the cycle from the first selected face.
- 7 Clear the current selection by clicking outside the geometry.
- 8 Set the **Select All Adjacent** button at the top of the Visualization/Selection toolbar.

9 Click on vertex 5 again, and all the adjacent faces are selected.



Modeling Physics and Equations

This chapter explains a range of methods you can use when building models in COMSOL Multiphysics. Read this chapter to learn several approaches for specifying physics, how to access predefined variables and create your own, how to specify and display units, how to explore the model using the Model Tree, how to use coupling variables, and much more.

Covering how to set up a model, this chapter outlines different options for the preprocessing steps needed to define:

- Boundary and interface conditions
- Material properties or domain equations
- Initial conditions

For flexible specification of these model properties, a great number of variables and functions are available.

For examples of modeling different types of physics and complete model examples, see the *COMSOL Multiphysics Modeling Guide* and the *COMSOL Multiphysics Model Library*.

Variables and Expressions

This section describes the syntax and names for variables and expressions that you define or that are predefined in a COMSOL Multiphysics application mode or model.

Using Variables and Expressions

To specify a coefficient or a material property, you can type a value or an expression directly in the corresponding edit field. Such an expression can contain:

- Numerical values
- Mathematical and logical functions
- Space coordinates, time, and the dependent variables in any application mode in the model as well as their space derivatives and time derivatives
- *Application mode variables*, which are predefined specifically for each application mode
- Other predefined variables
- Constants, expressions, coupling variables, and interpolation functions you have defined
- Units
- COMSOL Multiphysics functions and operators such as `diff` and `flc1hs`
- COMSOL Script functions
- MATLAB functions (only with the MATLAB interface)

You can use these types of variables and expressions in all the dialog boxes on the **Physics** menu for point settings, edge settings, boundary settings, and subdomain settings.

Variable Classification and Geometric Scope

COMSOL Multiphysics provides a set of variables you can use in expressions when specifying a model and for visualizing and postprocessing the solution. A number of variables are common to all application modes, one example being space coordinate variables such as x , y , and z . In addition, each application mode has its own set of variables that represent quantities relevant for the physics that it covers. This section describes the available variables and their naming and meanings.

Variables are divided into the following categories:

- Constants (variables that are independent of the geometry)
- Geometric variables (variables that characterize geometric properties)
- Field variables (dependent variables and variables derived from them)
- Special variables (variables for time, eigenvalues, and phasors)

Constants and geometric variables are always available. The choice of application mode and the dimension of the domains in the model affect the set of available field variables and special variables. Equations can be active in different domains, which also affects the set of available variables. Equation variables corresponding to certain equation terms are available only in the particular domains (such as boundaries, subdomains, points) where the equation is active. Equation variables defined on domains of a dimension lower than the subdomain are active if the adjacent subdomain is active.

Defining Constants, Expressions, and Coupling Variables

Defining constants and variables simplifies model specification and postprocessing. To add constants and variables, use the appropriate commands on the **Options** menu. In all dialog boxes, you can change the width of the columns for definition and display of constants and variables by dragging the dividing lines between columns at the columns headers.

CONSTANTS

Constants are global, that is, they are the same for all geometries and subdomains. Constants can depend on other constants and contain any mathematical functions. Use constants in any physics settings, expressions, and coupling variables. You can also use them in expressions while postprocessing the solution.

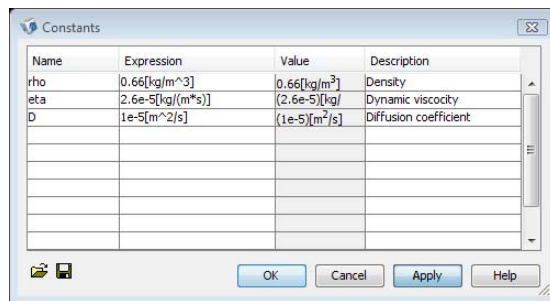


Figure 4-1: The Constants dialog box.

Adding Constants

You can add as many constants as desired:

- 1 In **Options** on the Main menu, choose **Constants**.
- 2 In the resulting **Constants** dialog box make an entry in the **Name** edit field.
- 3 Type the expression defining the value of the constant in the **Expression** edit field. This expression can include any constants already defined and also units. Click **Apply** or click in another edit field to evaluate the constant. The numeric value (and unit in the model's base unit system, if applicable) appears in the **Value** column.
- 4 Add a description of the constant in the **Description** edit field. This optional description can be any string.
- 5 Add more constants by entering values in other rows in the table.

You can modify a constant by editing its expression.

You can resize the **Constants** dialog box by dragging it at any of the dialog box's edges, which is useful if a model contains many constants.

Saving and Loading Constants

You can save the constants to a text file and also load constants from a text file into the **Constants** dialog box.

To save the constants to a file, click the **Save** button in the lower-left corner of the dialog box. COMSOL Multiphysics saves the information in three tab-separated columns to a text file that you define in the **Export Variables** dialog box. In each row, the first column contains the name of the constant, the second column contains the expression that defines its value, and the third column contains the description.

You can also load constants from a text file where each row—separated by tabs or colons—contains the name, expression, and description. Click the **Open** button in the lower left corner of the **Constants** dialog box. This opens the **Import Variables** dialog box, where you can browse for text files. COMSOL Multiphysics adds the constants to the end of the current list of constants in the dialog box.

Deleting a Constant

To delete a constant:

- 1 Clear the **Name** edit field
- 2 Click **Apply**.

EXPRESSION VARIABLES

Add symbolic *expression variables* or *expressions* using the dialog boxes that you open from the **Expressions** submenu on the **Options** menu. *Global expressions* are available globally in the model, and *scalar expressions* are defined the same anywhere in the current geometry. With *boundary expressions*, *subdomain expressions*, *point expressions*, and *interior mesh boundary expressions* you can also create expressions that have different meanings in different parts of the model. Expressions can contain other variables and expressions that COMSOL Multiphysics computes when solving the model, such as the solution components, other expression variables, the space variables, and time.

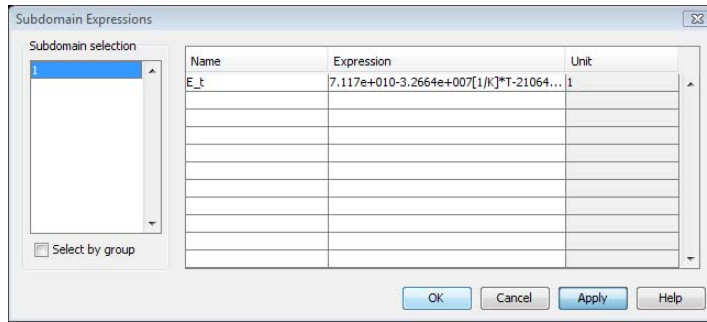


Figure 4-2: The Subdomain Expressions dialog box.

Expression variables can make a model easier to understand by introducing short names for complicated expressions. Another use for expression variables is during postprocessing. If you need to view a field variable throughout the model, but it has different names in different domains, create an expression variable made up of the different domains and then plot that variable. The plot then shows the entire field in all domains, eliminating the need to plot one variable at a time and overlay the plots.

Adding an Expression Variable

Follow these steps to add or edit an expression variable:

- 1 Go to the **Options** menu, point to **Expressions**, and then click:
 - **Global Expressions** to define expression variables that are valid everywhere in the model, including additional states for ODEs, algebraic equations, or other global equations that you specify in the **Global Equations** dialog box
 - **Scalar Expressions** to define scalar expression variables that are valid on all geometry levels everywhere in the current geometry
 - **Subdomain Expressions** to define expression variables for use on subdomains
 - **Boundary Expressions** to define expression variables for use on boundaries
 - **Edge Expressions** (appears only in 3D models) to define expression variables for use on edges
 - **Point Expressions** to define expression variables for use on points
 - **Interior Mesh Boundary Expressions** to define expression variables for use on interior mesh boundaries (see “Using Equation Contributions on Interior Mesh Boundaries” on page 280 in the *COMSOL Multiphysics Modeling Guide*)
- 2 Select the geometry domains where the expression variable exists (this step does not apply to global expressions and scalar expressions). A variable can use different expressions on different domains.
- 3 Make an appropriate entry in the **Name** edit field.
- 4 Make an entry in the **Expression** field to define the variable in all the domains where it exists.
- 5 For global expressions and scalar expressions, add a description of the variable in the **Description** edit field. This step is optional.
- 6 Click **Apply**.

To remove a global expression variable or a scalar expression variable, clear the **Name** edit field and click **Apply**.

Saving and Loading Global Expression Variables and Scalar Expression Variables

You can save global expression variables and scalar expression variables to text files and also load such variables from text files into the **Global Expressions** and **Scalar Expressions** dialog boxes. This works in the same way as for constants; see “Saving and Loading Constants” on page 140 for more information.

COUPLING VARIABLES

Coupling variables are powerful options that can help you accomplish a number of important modeling goals. A generalization of ordinary expressions, coupling variables can define nonlocal couplings including mesh transformations, integrals over

subdomains, and projections. The coupling variables are also the foundation of the *extended multiphysics* capabilities in COMSOL Multiphysics. With coupling variables you can take information from one type of model domain (a boundary, for example) and make it available on other model domains (a subdomain, for example). For more information on coupling variables, see the section “Using Coupling Variables” on page 269.

The definitions of expression variables and coupling variables are based on a geometry (except for scalar variables). To create such variables, make sure a geometry is already prescribed for the model.

MOVING AND CLEARING ROWS IN TABLES FOR VARIABLES

In tables for variables and constants you can move rows up and down and also clear rows. This makes it possible to rearrange their order after you have created them. To do so:

- 1 In the table, select the row you wish to move or clear.
- 2 Right-click to open a shortcut menu.
- 3 Click **Move Up** on the shortcut menu to move the row up one step; click **Move Down** to move the row down one step; click **Clear** to remove the contents of the row.

This feature is available in the following dialog boxes:

- The **Constants** dialog box
- Dialog boxes for expression variables
- Dialog boxes for coupling variables
- The table containing application mode variables in the physics settings dialog boxes for the equation system that you reach by going to the **Physics** menu and pointing to **Equation System**.

Note: It is not possible to clear the contents of rows with application mode variables.

Specifying Varying Coefficients and Material Properties

When assigning varying material properties, for example, you can enter expressions that use the space variables, time, results from other application modes in the model, results from the current application mode, interpolation from gridded data that you

enter into a table or have stored in a file, and interpolation from measured data. For more information on using interpolation to specify spatially varying and nonlinear material properties, see “Interpolation of External Data and Nonlinear Materials” on page 236.

The types of properties listed in the following sections are some of the most common for coefficients and material properties that vary during an analysis:

SPACE-VARYING PROPERTIES

Model properties can vary in space for all analysis types and solvers. When defining these properties, use the names of the space variables (independent variables), which are normally x , y , and z . The space variables in axisymmetric models are normally called r and z . You can also use space variables in expression variables and coupling variables. For example, you might type an expression such as $K0+0.1*x$ in an edit field if thermal conductivity varies with x . You also need to use space variables when using interpolation commands with gridded and measured data sets. It is possible to change the names of the space variables when creating a new geometry and later refer to those names when specifying a property.

TIME-VARYING PROPERTIES

Model properties can vary in time for time-dependent analyses. When defining properties that vary in time, use the reserved variable name for time, t (see “The Time Variable” on page 181 for details).

NONLINEAR PROPERTIES

In a nonlinear model, the model properties depend on the variables for which you solve (the dependent variables). For example, you might enter an expression like $K0+0.1*T^2$ in an edit field if thermal conductivity varies quadratically with temperature T . COMSOL Multiphysics automatically detects if a model is nonlinear and then uses a nonlinear solver. For a more exact definition of a nonlinear model, see “Which Models are Nonlinear?” on page 381.

See also “Interpolation of External Data and Nonlinear Materials” on page 236 for insight on how to specify material properties that vary with the dependent variable according to a trend in data.

Copying Physics Settings

You can copy and paste point, edge, boundary, and subdomain settings using the copy and paste commands from the **Edit** menu or the equivalent shortcut keys. It is also

possible to copy the settings for coupling variables and point, edge, boundary, and subdomain expressions. COMSOL Multiphysics can copy the contents of the clipboard to multiple domains of the same type. The copy and paste functionality differs slightly depending on whether the dialog box for the physics setting is open or not:

- If the domain selection list in the dialog box has the input focus (being the location where the user is currently directing input), pressing Ctrl+C copies the physics settings from the dialog box to the clipboard, regardless of whether the settings are applied or not. Pressing Ctrl+V when the domain selection list in the dialog box has the input focus copies the contents of the clipboard to the edit fields of the dialog box; that is, the settings are not automatically applied.
- Otherwise, pressing Ctrl+C copies the physics settings for the currently selected application mode to the clipboard, and pressing Ctrl+V directly applies the contents of the clipboard to the selected domains.

The contents of the clipboard are preserved during a COMSOL Multiphysics session. This makes it possible to copy and paste model properties from one model to another.

You can also copy physics settings using the Model Tree (see “Using the Model Tree” on page 223).

Entering Vector-Valued Expressions

You can enter vector-valued expressions such as extra grid-line coordinates using the following formats:

- A space-separated list of values: `10 15 23 29.7 30`.
- Equally spaced values using the colon operator as in `start value:step:end value`. For instance, `0:0.2:3` creates the values 0, 0.2, 0.4, ..., 2.6, 2.8, 3.0. The step size is 1 if you provide only start and end values and skip the step value.
- Equally spaced values using the `linspace` function. For instance, `linspace(0,100,15)` creates a vector of 15 equally spaced values from 0 to 100. If you provide only start and stop values and do not indicate the number of values, `linspace` provides a vector of 100 equally spaced values.
- Logarithmically spaced values using the `logspace` function. For instance, `logspace(x1,x2,n)` generates a row vector of n logarithmically equally spaced values between decades 10^{x1} and 10^{x2} . If $x2$ equals π , the values fall between 10^{x1} and π .

You can combine these formats in one expression to create a vector of values that contains segments with linearly or logarithmically spaced values as well as other freely specified values.

The following modeling features support vector-valued expressions:

- Extra grid lines in the **Axes/Grid Settings** dialog box
- Line and point coordinates when using the **Line** and **Point** dialog boxes
- The times for output from the time-dependent solver and the list of parameter values for the parametric solvers in the **Solver Parameter** dialog box
- The contour levels, the streamline start-point coordinates, and the coordinates in arrow plots. These visualization settings appear in the **Plot Parameters** dialog box. Whenever you specify a number of coordinates in postprocessing dialog boxes, COMSOL Multiphysics uses scalar expansion—if one component is the same for all coordinates, you can enter a single number in the corresponding text field. For example, to get 100 linearly spaced points between $y = 6$ and $y = 7$ along $x = 3$, enter it as the single scalar 3 for x and then `linspace(6,7,100)` for y . Thus, you need not enter `linspace(3,3,100)` for x .
- The edge-vertex distribution on boundary segments in the **Mapped Mesh Parameters** dialog box.
- The element-layer distribution in the **Extrude Mesh** and **Revolve Mesh** dialog boxes.

Note: For these settings you have access both to the element-by-element operators `.*`, `./`, and `.^` as well as the matrix operators `*`, `/`, and `^` (multiply, divide, and power operators, respectively). You typically use the element-by-element operators for an expression such as `[0:1/20:1].^2`. (The matrix power operator does not work in this case because the matrix is not square.)

Using Mathematical and Logical Operators and Functions

You can use many common mathematical and logical operators and functions along with constants to specify coefficients and material properties. The following tables show the available functions and operators:

TABLE 4-1: UNARY OPERATORS

OPERATOR	DESCRIPTION
+	unary plus
-	unary minus
~	logical not

TABLE 4-2: BINARY OPERATORS

OPERATOR	DESCRIPTION
+	plus
-	minus
*	multiply
/	divide
^	power
==	equal
~=	not equal
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
	or
&	and

TABLE 4-3: MATHEMATICAL FUNCTIONS AND CONSTANTS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
abs	absolute value	abs(x)
acos	inverse cosine	acos(x)
acosh	inverse hyperbolic cosine	acosh(x)
acot	inverse cotangent	acot(x)
acoth	inverse hyperbolic cotangent	acoth(x)
acsc	inverse cosecant	acsc(x)
acsch	inverse hyperbolic cosecant	acsch(x)

TABLE 4-3: MATHEMATICAL FUNCTIONS AND CONSTANTS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
angle	phase angle	angle(x)
asec	inverse secant	asec(x)
asech	inverse hyperbolic secant	asech(x)
asin	inverse sine	asin(x)
asinh	inverse hyperbolic sine	asinh(x)
atan	inverse tangent	atan(x)
atan2	four-quadrant inverse tangent	atan2(y, x)
atanh	inverse hyperbolic tangent	atanh(x)
besselj	Bessel function of the first kind	besselj(a, x)
bessely	Bessel function of the second kind	bessely(a, x)
besseli	modified Bessel function of the first kind	besseli(a, x)
besselk	modified Bessel function of the second kind	besselk(a, x)
conj	complex conjugate	conj(x)
cos	cosine	cos(x)
cosh	hyperbolic cosine	cosh(x)
cot	cotangent	cot(x)
coth	hyperbolic cotangent	coth(x)
csc	cosecant	csc(x)
csch	hyperbolic cosecant	csch(x)
eps	floating point relative accuracy	eps
erf	error function	erf(x)
exp	exponential	exp(x)
i, j	imaginary unit	i
imag	imaginary part	imag(u)
inf	infinity	inf
log	natural logarithm	log(x)
log10	common logarithm (base 10)	log10(x)
log2	base 2 logarithm	log2(x)
max	maximum of two arguments	max(a, b)
min	minimum of two arguments	min(a, b)
mod	modulo operator	mod(a, b)

TABLE 4-3: MATHEMATICAL FUNCTIONS AND CONSTANTS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
NaN, nan	not-a-number	nan
real	real part	real(u)
pi	pi	pi
sec	secant	sec(x)
sech	hyperbolic secant	sech(x)
sign	signum function	sign(u)
sin	sine	sin(x)
sinh	hyperbolic sine	sinh(x)
sqrt	square root	sqrt(x)
tan	tangent	tan(x)
tanh	hyperbolic tangent	tanh(x)

Specifying Discontinuous Functions

To specify a discontinuous function, such as a step in space or time, you can use logical functions that evaluate to 1 if true and 0 otherwise. For instance, the function

$$\sin(2*\pi*t)*(t<10)$$

defines a sine wave that exists for times until 10 seconds and afterward takes on the value of 0.

If a coefficient or a material property contains a step function or some other discontinuity, convergence problems can arise. For time-dependent problems, the time-stepping algorithm can run into problems. For stationary problems, mesh-resolution issues can arise such as overshooting and undershooting of the solution due to infinite flux problems.

To avoid problems with a discontinuity, try replacing it with a smoothed switch function that emulates steps. Doing so serves two purposes:

- Numerical reliability and convergence are improved.
- What you think of as a step function is, in reality, a smoothed continuous function because of inertia.

The following smoothed functions are available:

- `f1smhs`, a smoothed Heaviside function with a continuous first derivative and overshoot on both sides of the step. The overshoot ensures that the integral from 0

to infinity is correct. $y=f1smhs(x, scale)$ approximates the logical expression $y = (x>0)$ by smoothing the transition within the interval $-scale < x < scale$. $f1dsmhs$ is the derivative of the smoothed Heaviside function.

- $f1smhsign$, a smoothed sign function with a continuous first derivative. $y = f1smhsign(x, scale)$ approximates the function $y = sign(x)$ by smoothing the transition within the interval $-scale < x < scale$. $f1dsmhsign$ is the derivative of the smoothed sign function.
- $f1c1hs$, a smoothed Heaviside function with a continuous first derivative without overshoot. Its syntax is similar to the functions just described.
- $f1c2hs$, a smoothed Heaviside function with a continuous second derivative without overshoot. Its syntax is similar to the functions just described.

In the interval $-scale < x < scale$, the functions $f1smhs$ and $f1smhsign$ are defined by a 7th-degree polynomial chosen so that the 2nd derivative is continuous. Moreover, the moments of order 0, 1, and 2 agree with those for the Heaviside function and the sign function, respectively. This implies that the functions have small overshoots.

Now consider an example. You can use $f1c1hs$ to model the heat capacity C_p of a phase-changing material. Assume that a crystalline material has a heat capacity of 1 kJ/kg. Its melting point at the present pressure is 273 K. The liquid phase has a heat capacity of 2 kJ/kg. Run the following commands in COMSOL Script or in MATLAB to generate this behavior and plot C_p around the melting point:

```
scale = 0.1;  
T = [272.5:.01:273.5];  
cp = 1+f1c1hs(T-273,scale);  
plot(T,cp); axis([272 274 .8 2.1])
```

Using Analytic Functions

Just as you can define variables, you can define analytic functions. They extend variables with the ability to bind parameters during function calls. In other words, you need not know the actual names of the variables in an expression when you write the function. You define analytic functions in the **Functions** dialog box.

Assume you want to define a function $fun(a, b, c, d)$, which returns $a+b*\cos(c)$.

- 1 From the **Options** menu, choose **Functions**.
- 2 Click the **New** button.
- 3 In the **New Function** dialog box that appears, click the **Analytic** button and type fun in the **Function name** edit field. Click **OK** to return to the **Functions** dialog box.

- 4 In the **Arguments** edit field, type a, b, c, d .
- 5 In the **Expression** edit field, type $a+b*\cos(c)$.
- 6 Click the **Manual** button in the **Derivatives** area, then type $1, \cos(c), \text{diff}(a+b*\cos(c), c), 0$ in the associated edit field.
- 7 Click **OK**.

COMSOL Multiphysics must know the derivative of a function if you are using a variable that depends on the solution in a function argument. To specify which derivatives to use, click the **Manual** button and enter the derivatives in the associated edit field. Notice the use of the `diff` operator (see “Using Special Operators” on page 163 for more information). If you click the **Auto** button in the **Derivatives** area, COMSOL Multiphysics uses the `diff` operator on the expression to get the derivatives. Select the **May produce complex output for real arguments** check box if the defined function works similarly to `sqrt`, that is, it sometimes returns complex values for a real-valued input.

Piecewise Analytic Functions

To define piecewise analytic functions, use the **Functions** dialog box. To define a new piecewise analytic function, do the following steps:

- 1 From the **Options** menu, choose **Functions**.
- 2 Click the **New** button.

- 3 In the **New Function** dialog box that appears, type the name of the function in the **Function name** edit field, then click the **Piecewise analytic** option button, and finally click **OK**.

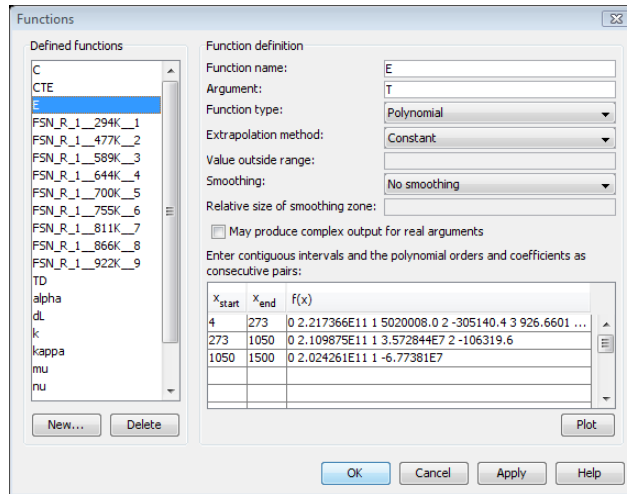


Figure 4-3: The Functions dialog box, used here to edit a piecewise analytic function defining the Young's modulus for the material 1010 (GI0100) in the Material Library.

You choose between three different function types in the **Function type** list:

- **Polynomial**
- **Exponential polynomial**
- **General**

POLYNOMIAL

Use the Polynomial function type to define functions using different polynomials on the intervals, that is:

$$f(x) = \sum_i a_{ij} x^i \quad x_{j\text{start}} < x \leq x_{j\text{end}}$$

where i can be any integer, also negative and zero.

EXPONENTIAL POLYNOMIAL

Use the Exponential polynomial function type to define functions using different exponential functions with a polynomial exponent on the intervals, that is

$$f(x) = \sum a_{ij}x^i \quad x_{j\text{start}} < x \leq x_{j\text{end}}$$

where i can be any non-negative integer.

GENERAL

Use the General function type to create functions that use an analytic expression on the different intervals. For the **General** type you also need to specify the name of your argument in the **Argument** edit field.

Also specify how COMSOL Multiphysics should treat arguments that fall outside the intervals. There are four extrapolation methods to choose from in the **Extrapolation method** list:

- **No extrapolation**

The function is not defined outside the interval. Trying to evaluate the function generates an error and evaluates to NaN (Not-a-Number).

- **Constant** (default)

The extrapolation method uses the value from the start point of the first and the end point of the last interval on the corresponding sides.

- **Nearest function**

The function at the first or last interval (depending of the side) is used to evaluate at the actual point where a value is requested.

- **Specific number**

If selecting the specific-number method you can assign a single value (usually zero or NaN) to all points outside the intervals. Enter the single value in the **Value outside range** edit field.

The functions defined in the different intervals might create a discontinuous function, which can cause problems for the solvers. One way to avoid this is to smooth the function at the interval borders. You choose between four different levels of smoothing in the **Smoothing** list:

- **No smoothing** (default)

- **Continuous function**

- **Continuous first derivative**

- **Continuous second derivative**

Specify the size of the smoothing zone in the **Relative size of smoothing zone** edit field. Relative size meaning in relation to the size of the intervals on both sides of the border.

Select the **May produce complex output for real arguments** check box if the defined function works similarly to `sqrt`; that is, it sometimes returns complex values for a real-valued input.

You enter the start and end points together with the definition of the function on the interval directly into a table. The table has three columns:

- x_{start}
- x_{end}
- $f(x)$

Specify the interval limits in the x_{start} and x_{end} columns.

Note: The intervals need to be contiguous and in ascending order.

Specify the function definition in the $f(x)$ column. You define the function differently depending of the **Function type**. The Polynomial and Exponential polynomial function types are defined using consecutive pairs specifying the polynomial order and the corresponding coefficient. For example, to specify the function

$$f(x) = 1.23 + 1.2 \cdot 10^3 x^3 - 6.58 x^5$$

enter 0 1.23 3 1.2e3 5 -6.58 in the $f(x)$ column. You specify a General function by simply entering the expression using the argument you have specified in the **Argument** edit field. Enter the above function as `1.23+1.2e3*x^3-6.58*x^5` if you specify x as the argument.

You get a preview of your function by clicking the **Plot** button.

Using Functions Based on Interpolated Data

To define functions based on interpolated data, use the **Functions** dialog box, which you open from the **Options** menu. The interpolation data can be both structured, (defined on a grid) or unstructured (defined on a generic point cloud). To define a new interpolation function, click the **New** button to open the **New Function** dialog box. Specify a name for the function, then click the **Interpolation** button. Next choose a method for entering data (a user-defined table or data from a file or the COMSOL

Script or MATLAB workspace) and, if applicable, a data source, then click **OK**. For file data, select the **Store data in model** check box to load the data into the model and make it independent of the original data file.

Note: If the file contains a large amount of data, selecting the **Store data in model** check box significantly increases the size of the Model MPH-file and Model M-file.

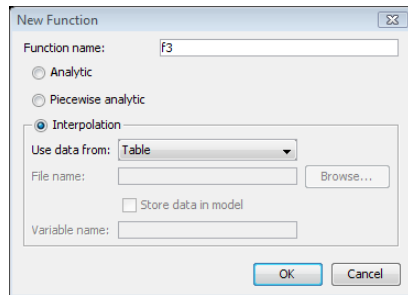


Figure 4-4: The New Function dialog box, used here to specify an interpolation function.

For functions of one variable, you can choose between the following interpolation methods:

- Nearest neighbor
- Linear
- Piecewise cubic
- Cubic spline

For functions of more than one variable COMSOL Multiphysics supports only the nearest-neighbor and linear interpolation methods. Piecewise-cubic interpolation is a method using a piecewise-cubic Hermite polynomial with continuous first derivatives. It preserves the shape of the data and respects monotonicity. The cubic-spline method also performs interpolation with a piecewise cubic polynomial. Here, even second derivatives are continuous; however, the method does not necessarily respect monotonicity.

Also specify how COMSOL Multiphysics treats arguments that fall outside the grid or mesh of points. There are four extrapolation methods to choose from for structured interpolation:

- Constant

- Interpolation function
- Linear
- Specific number

Unstructured interpolation supports using a constant or a specific number only.

Constant means that the extrapolation method uses the value from the closest point inside the grid (for structured interpolation) or the value from the closest mesh element (for unstructured interpolation). The interpolation function evaluates the polynomial from the closest grid point at the actual point where a value is requested. Linear means that the function is linear outside the grid with a continuous function value and continuous first derivative at the boundary of the grid. This extrapolation method can only be used together with the interpolation methods piecewise cubic and cubic spline. By selecting the specific number option, you can assign a single value, usually zero or NaN, to all points outside the grid or mesh.

The **Use space coordinates as default function arguments** check box, when selected, makes it possible to use the space coordinates as default function arguments. Thus, if the space coordinates are x , y , and z , selecting this check box for a function `myfun` allows you to write `myfun` instead of `myfun(x, y, z)` when using the function. You can use this option only for geometries of the same dimension as the function. The complete version (`myfun(arg1, arg2, arg3)`) is always available.

For functions of one variable you can enter the data directly into a lookup table (see Figure 4-5).

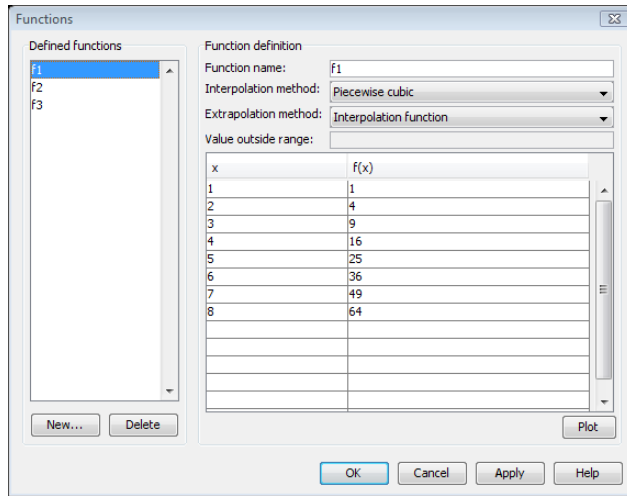


Figure 4-5: The Functions dialog box, used here to edit an interpolation function.

To get a graph of the interpolation function, click the **Plot** button. The plot includes extrapolation (the red dashed lines outside of the interval for x) to show how the extrapolation method works (see Figure 4-6).

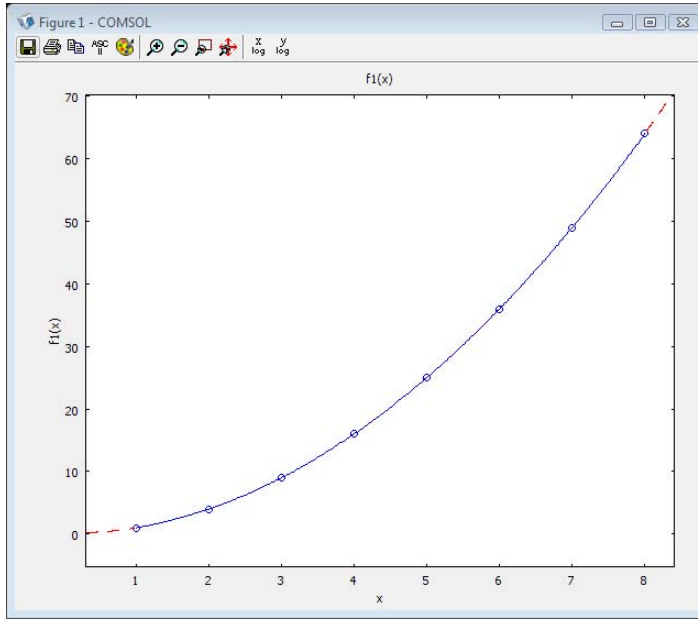


Figure 4-6: A plot of the function $f1(x)$. Notice the dashed red lines that indicate the extrapolation.

For functions of one to three variables, you can either retrieve data from a text file, COMSOL Script, or MATLAB. The interpolation functions support the three file formats that can be exported from **Export>Postprocessing Data** on the **File** menu: the **Grid, data** format on a regular grid, the **Coordinates, data** format, and the **Nodes, elements data** format, described in more detail in “Exporting Postprocessing Data to a File” on page 487.

When the data is stored in a text file, the function normally only stores the name of the text file and reads data from the file during the analysis. Select the **Store data in model** check box when creating the function to load the data into the model and make it independent of the original file.

- **Nodes, elements, data:**

```
%Coordinates
One to three columns containing x, y (optional) and
z (optional)
%Elements
Triangulation where each row contains the row indices of the
points in the Coordinates section that make up one element
```



```
(triangular in 2D, tetrahedral in 3D)
%Data (funname)
Column of data values for each point
```

It is possible to include more than one function in the file as long as a %Data header separates them one from the other.

- **Grid, data:**

```
% Grid
x grid points separated by spaces
y grid points separated by spaces (optional)
z grid points separated by spaces (optional)
% Data
Data values separated by spaces
```

Each row contains values for different x grid points for fixed values of y and z . The rows first increase the y grid value and then the z grid value. The grid points can also represent another independent variable that the data values depend on. For example, the “grid points” can be temperature values and the data values the thermal conductivity at these temperatures (see the example in “Interpolation of External Data and Nonlinear Materials” on page 236). 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. You can use the `postwriteinterpfile` function to create a file of this format from COMSOL Script or MATLAB.

It is possible to include more than one function in the file as long as a %Data header separates them one from the other.

- **Coordinates, data:**

```
%Header (optional)
Columns containing x, y (optional) and z (optional) followed
by data columns.
```

You can include function names in the header. In that case, the input columns must be labeled with x , y , and z , respectively, depending on input dimension. For example, a file with the following content creates two 2D functions named `myfun1` and `myfun2`:

```
% x y myfun1 myfun2
0 0 0.12 0.34
0 1 0.52 1.50
1 0 0.67 0.91
```

If the file does not include any header to indicate the function dimension, the software assumes that it is identical to the largest geometry dimension present in the model. A file with four columns, for example, is interpreted as one 3D

function in a 3D model, two 2D functions in a 2D model, and three 1D functions in a 1D model.

Note: To use the function names supplied in the file (in the cases of the **Nodes**, **elements**, **data** and **Coordinates**, **data** formats), leave the **Function name** edit field empty when creating a new function (see Figure 4-4 on page 155).

To retrieve data from COMSOL Script or MATLAB, create a structure on the command line with the fields `x`, `y` (optional), `z` (optional), `data`, and `elem` (optional). For structured data, `x`, `y`, and `z` are vectors specifying the grid, and `data` is a multidimensional array of the same format as produced by the function call `[xx,yy,zz] = meshgrid(x,y,z)`; . For example, to define the function $f(x,y,z) = xysin(z)$ on a unit-cube grid and prepare it for export to COMSOL Multiphysics, type the following on the COMSOL Script or MATLAB command line:

```
x = linspace(0,1,11); y = linspace(0,1,11); z = linspace(0,1,11);
[xx,yy,zz] = meshgrid(x,y,z);
data = xx.*yy.*sin(zz);
f.x = x; f.y = y; f.z = z; f.data = data;
```

When COMSOL Multiphysics evaluates the function, the software performs a structured interpolation from the data values on the grid to the coordinates where the function is evaluated. See “A Rock Fracture Flow Model” on page 289 of the *COMSOL Multiphysics Model Library* for an example using data interpolation.

For unstructured data, `x`, `y` (optional), `z` (optional), and `data` must be the same length. `elem`, if present, must contain a valid triangulation of the points such that each column contains the indices of the points describing one mesh element. For example, to define the function $f(x,y,z) = xysin(z)$ on a random point cloud, type the following on the COMSOL Script or MATLAB command line:

```
x = rand(1,20); y = rand(1,20); z = rand(1,20);
data = x.*y.*sin(z);
elem = delaunay3(x,y,z)';
f.x = x; f.y = y; f.z = z; f.data = data; f.elem = elem;
```

COMSOL Multiphysics stores the data that it retrieves from COMSOL Script or MATLAB in the function when it is created, and the connection to the original structure is lost.

Using COMSOL Script or MATLAB Functions

When running COMSOL Multiphysics with COMSOL Script or MATLAB, you can use command-line functions in a COMSOL Multiphysics expression. In fact, when COMSOL Multiphysics encounters an unknown function it assumes that it is a COMSOL Script (or MATLAB) function and searches for the M-file defining the function in the current directory. If the M-file is not found there, COMSOL Multiphysics searches the directories in the current path.

To inspect the current directory and the current path, open COMSOL Script from the **File** menu, and then type `pwd` and `path`, respectively, on the command line. These settings are shared between COMSOL Multiphysics and COMSOL Script, so you modify them in both environments simultaneously using the commands `cd` and `path` on the COMSOL Script command line.

COMSOL Multiphysics assumes that an unknown function's derivatives with respect to its parameters are zero, and that the function returns real values for real input arguments. You can use the **Functions** dialog box to change these defaults.

Assume, for instance, that you also want to use COMSOL Script's or MATLAB's Bessel function of the third kind (Hankel function) in COMSOL Multiphysics. Open the **Functions** dialog box from the **Options** menu and create the following analytic functions (see "Using Analytic Functions" on page 150):

FUNCTION NAME	ARGUMENTS	EXPRESSION	DERIVATIVES
<code>besselh</code>	<code>nu, x</code>	<code>besselh(nu, x)</code>	<code>0, (besselh(nu-1, x) -besselh(nu+1, x))/2</code>
<code>bessel</code>	<code>x</code>	<code>besselh(1, x)</code>	<code>(besselh(0, x) -besselh(2, x))/2</code>

The first function introduces the `besselh` function to COMSOL Multiphysics and provides its derivatives. The second function sets up a shorthand called `bessel` for `besselh(1, x)`.

Function Input/Output Considerations

The functions you use must support matrix arguments of at least three dimensions. All common functions such as `sin`, `abs`, and other mathematical functions support matrix arguments.

When you write your own functions for specifying inhomogeneous materials, logical expressions, time-dependent sources, or other model properties, keep in mind that the

input arguments can be matrices. The output must have the same format as the input. Consider the following example function where the coefficient c depends on the x coordinate:

```
function c = func1(x)
if x > 0.6
    c = x/1.6;
else
    c = x^2+0.3;
end
```

The above function looks good at a first glance, but it does not work in COMSOL Multiphysics because the input x is a matrix.

- First of all, you must use element-by-element multiplication, division, and power, that is, the operators `.*`, `./`, and `.^`. Replace expressions such as `x/1.6` and `x^2+0.3` with `x./1.6` and `x.^2+0.3`, respectively.
- The comparison `x > 0.6` returns a matrix with ones (true) for the entries where the expression holds true and zeros (false) where it is false. The function evaluates the conditional statement only if all the entries are true (1).

You can replace the `if` statement with a single assignment to the indices retrieved from the `x > 0.6` operation and another assignment to the indices where `x ≤ 0.6`. All in all, the function could look like this:

```
function c = func2(x)
c = (x./1.6).*(x>0.6) + (x.^2+0.3).*(x<=0.6);
```

Sometimes the calculations involve functions that do not support general matrix inputs. It is then necessary to reshape the input arguments into vector form before doing the calculations. It is important to reshape the output back into the original form afterward. The following is an example of this idea:

```
function y = mypolyval(p,x)
sx = size(x);
x = x(:);
y = polyval(p,x);
y = reshape(y,sx);
```

This function wraps the original `polyval` command so that it works with the input from COMSOL Multiphysics.

Using Special Operators

COMSOL Multiphysics includes special operators that are available for modeling and postprocessing. Many application modes and equation forms use these operators to implement equations and special functionality. Table 4-4 summarizes these operators:

TABLE 4-4: SPECIAL OPERATORS

OPERATOR	DESCRIPTION
<code>diff(f, x)</code>	Differentiation operator: Differentiation of f with respect to x .
<code>pdiff(f, x)</code>	Differentiation operator: Differentiation of f with respect to x . No chain rule for dependent variables.
<code>test(expr)</code>	Test function operator.
<code>nojac(expr)</code>	No contribution to the Jacobian.
<code>up(expr)</code>	Evaluate expression as defined in adjacent upside.
<code>down(expr)</code>	Evaluate expression as defined in adjacent downside.
<code>mean(expr)</code>	Mean value of expression as evaluated on adjacent boundaries.
<code>depends(expr)</code>	True if expression depends on the solution.
<code>islinear(expr)</code>	True if expression is a linear function of the solution.
<code>dest(expr)</code>	Evaluate parts of an integration coupling expression on destination side.
<code>if(cond, expr1, expr2)</code>	Conditional expression evaluating the second or third argument depending on the value of the condition.
<code>quad(f, x, a, b, tol)</code>	Adaptive numerical quadrature of f with respect to x .
<code>with</code>	Access any solution during postprocessing.
<code>at</code>	Access the solution at any time during postprocessing.
<code>ppr</code>	Accurate derivative recovery.
<code>pprint</code>	Accurate derivative recovery within each domain group.
<code>reactf</code>	Accurate integrals of reaction forces and fluxes.
<code>adj(expr)</code>	Evaluate expression using the adjoint sensitivity.
<code>fsens(expr)</code>	Evaluate expression using the functional sensitivity.
<code>sens(expr, i)</code>	Evaluate expression using the forward sensitivity for the parameter given by the second argument.
<code>realdot(a, b)</code>	Treat complex numbers a and b as real 2-vectors and return their dot product.

THE DIFFERENTIATION OPERATORS

Use the `diff` operator to differentiate a variable with respect to another variable. For example, `diff(T,x)` means differentiation of T with respect to x . You can use the `diff` operator in the model settings. If it contains any of the dependent variables, it typically makes the model nonlinear. It is also possible to use the `diff` operator during postprocessing to visualize and compute derivatives. Some of the space derivatives are also available using predefined variables. For example, `uxx`, `diff(ux,x)`, and `diff(diff(u,x),x)` are equivalent. The `pdiff` operator works in a similar way but does not apply the chain rule for dependent variables. The expression `diff(E,TIME)` computes the reference time derivative of the expression E (see “Derivatives” on page 456 of the *COMSOL Multiphysics Modeling Guide*).

Examples of Using the Differentiation Operators

The expressions `diff(u^2,u)` and `pdiff(u^2,u)` both equal $2*u$. Note that `diff` also takes the space and time variables into account and treats their derivatives accordingly. In other words, if u is the dependent variable and x and t are the space coordinate and time, respectively, then `diff(u+x,x)` equals u_x+1 (u_x is the space derivative of u with respect to x), while `pdiff(u+x,x)` equals 1, because u is considered to be independent of anything but itself in the case of `pdiff`. Equivalently, `diff(u,t)` equals u_t , while `pdiff(u,t)` is zero.

THE TEST OPERATOR

The `test` operator is available for modeling using the weak formulation of the equations. The `test` operator creates the test function for the variable that it operates on. As an alternative to the operator `test(u)`, you can use `u_test`, that is, append `_test` to your dependent variable name. For an expression, `test(F(u,∇u))`, the test operator is equivalent to

$$\sum_i \text{test}(u_i) \frac{\partial}{\partial u_i} F(u_i, \nabla u_i) + \text{test}(\nabla u_i) \frac{\partial}{\partial \nabla u_i} F(u_i, \nabla u_i),$$

for all dependent variables u_i . For more information about test functions and variables for weak form modeling, see the chapter “The Weak Form” on page 349 of the *COMSOL Multiphysics Modeling Guide*.

THE NOJAC OPERATOR

The `nojac` operator makes sure that any expression that it operates on is excluded from the Jacobian computation. This is useful if a Jacobian contribution is not strictly necessary and the computational requirements for it are high. The k - ϵ turbulence

model in the *Chemical Engineering Module* is an application mode where the use of the `nojac` operator improves performance. For another example of its use, see the model “Process Control Using a PID Controller” on page 337 of the *COMSOL Multiphysics Model Library*.

THE UP, DOWN, AND MEAN OPERATORS

COMSOL Multiphysics can evaluate expressions on both sides of a domain. These operators are available only on boundaries. For an expression or a variable that is discontinuous across a boundary, the value is different on either side, and COMSOL Multiphysics normally displays the mean values on the boundary. Use the `up`, `down`, and `mean` operators to evaluate an expression on the upside or downside of the boundary or the mean value on the boundary. If the upside or downside is outside of the geometry, or if the variables in the expression are not active on that side, the `up` or `down` operator returns 0. For more information about the upside and downside of a boundary, see “Tangent and Normal Variables” on page 172.

THE DEPENDS OPERATOR

The `depends` operator returns 1 if the expression that it operates on depends on the solution; otherwise it returns 0. You can use this operator to check user-defined expressions for dependency on the solution.

THE ISLINEAR OPERATOR

The `isLinear` operator returns 1 if the expression that it operates on is a linear function of the solution; otherwise it returns 0. You can use this operator to check user-defined expressions for linearity with respect to the solution. The stationary solver does this automatically to choose between a linear or a nonlinear solver.

THE DEST OPERATOR

The `dest` (destination) operator is available for use in integration-coupling variable expressions. This operator forces the expression that it operates on to be evaluated on the destination points instead of the source points. This means that you can use the destination operator to create convolution integrals and other integral transforms. For instance, integrating the expression $u / ((\text{dest}(x) - x)^2 + (\text{dest}(y) - y)^2)$ gives the following function of x and y :

$$f(x, y) = \int \frac{u(x', y')}{(x - x')^2 + (y - y')^2} dx' dy'$$

For an example of using the `dest` operator for solving an integro-partial differential equation, see “An Integro-Partial Differential Equation” on page 132 in the *COMSOL Multiphysics Model Library*.

THE IF OPERATOR

The `if(cond, expr1, expr2)` operator implements a conditional expression. The first argument is a condition that COMSOL Multiphysics treats as a Boolean expression. If—at a particular evaluation point—`cond` is true, the second argument is evaluated, else the third argument is evaluated. That is, only one branch is evaluated at any evaluation point. Use the `if` operator to avoid illegal operations. For example, `if(x==0, 1, sin(x)/x)` is a valid and continuous expression for all values of x , including $x = 0$.

THE QUAD OPERATOR

The `quad` operator implements an adaptive recursive Simpson quadrature. Call `quad(f, x, a, b, tol)` to evaluate the integral of f with respect to x from a to b . The integration variable x must be a single variable name. The lower and upper limits, a and b , must be *scalar*, which in this case means that their values do not depend on the evaluation point. The final argument is a relative tolerance and must also be a scalar value. Note that `quad` is mainly intended for pre- and postprocessing of data and therefore does not give any contribution to the Jacobian if you use it in equations.

Because the limits must be scalar, it is sometimes necessary to transform the integral. For example, to integrate $f(x)$ with respect to x from 0 to u , where u is the dependent variable (not a scalar), introduce a new variable ξ such that $x = u\xi$. This changes the upper integration limit to 1 (a scalar) and the integration expression becomes `quad(f(xi*u)*u, xi, 0, 1, 1e-6)`.

THE WITH OPERATOR

Using the `with` operator, you can access any solution during postprocessing. For time-dependent problems, parametric problems, and eigenvalue problems, this makes it possible to use the solution at any of the time steps, the solution at any parameter value, or any eigensolution in a postprocessing expression. Use the solution number as the first input argument. The second input argument is the expression that you want to evaluate using this solution. For example,

```
with(3, u^2)
```

provides the square of the third eigensolution for an eigenvalue problem.

For an example using the `with` operator to verify that two eigensolutions are orthogonal, see the model “Isospectral Drums” on page 604 in the *COMSOL Multiphysics Model Library*.

THE AT OPERATOR

Using the `at` operator during postprocessing, you can access solution to a time-dependent problem at any time. COMSOL Multiphysics provides the solution at that time using interpolation. The first input argument is the time. The second input argument is the expression that you want to evaluate using this solution. For example,

```
at(12.5,u)
```

is the solution at 12.5 s.

You can combine the `at` operator with the `quad` operator to compute time integral transforms. For example, assume that the functions $f(t)$ and $g(t)$ are effectively zero outside the interval $[0, T]$, and that the corresponding time-dependent solution objects, `f` and `g`, are defined on a time interval that includes $[-T, T]$. Then, the time convolution integral

$$(f \bullet g)(t) \equiv \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$

is evaluated as

```
quad(at(tau,f)*at(t-tau,g),tau,0,T,tol)
```

where T and `tol` are scalar values for T and the relative tolerance (see page 166), respectively.

THE PPR AND PPRINT OPERATORS

When you apply the `ppr` operator on an expression, COMSOL Multiphysics uses polynomial-preserving recovery to evaluate all variables in the expression that are discretized using Lagrange shape functions; see “Accurate Derivative Recovery” on page 469. For example, if $e = ux + vy$, then

```
ppr(e^2)=(ppr(ux)+ppr(vy))^2
```

The `pprint` operator similarly applies polynomial-preserving recovery within each group of domains with equal settings. You can use these operators to get an estimate of the discretization error in the gradient, for example

```
ux-pprint(ux)
```

in a 1D model. Note that if you apply these operators on expressions that are used when solving the model, the software then computes the Jacobian approximately by ignoring the operator. For example, the Jacobian of `ux-pprint(ux)` is 0.

THE REACF OPERATOR

The reaction force operator (`reacf`) is available when computing integrals of reaction forces or fluxes during postprocessing. Apply the reaction force operator on the names of dependent variables in the **Boundary Integration** dialog box. For example, in structural mechanics, with dependent variable u and v corresponding to x - and y -displacements, use `reacf(u)` and `reacf(v)` in the **Boundary Integration** dialog box to access integrals of the reaction forces in the x - and y -direction, respectively. Access to the operator during postprocessing can be disabled by clearing the **Store reaction forces** check box on the **Output** page in the **Solver Manager** dialog box. The reaction force operator is not available when using weak constraints.

THE ADJ OPERATOR

When you apply the adjoint sensitivity operator (`adj`) to an expression, COMSOL Multiphysics uses the adjoint sensitivity solution instead of the primal solution for the evaluation. The adjoint sensitivity solution is available for postprocessing when the sensitivity solver has been used with the adjoint sensitivity method, and for the dependent variables that have been solved for.

THE FSSENS OPERATOR

When you apply the functional sensitivity operator (`fsens`) to an expression, the software uses the functional sensitivity solution for the evaluation. The functional sensitivity solution is available for postprocessing when the sensitivity solver has been used with either the adjoint or the forward sensitivity method, and for the sensitivity variables that have been solved for.

THE SENS OPERATOR

When you apply the forward sensitivity operator (`sens`) to an expression, the software uses the forward sensitivity solution instead of the primal solution for the evaluation. The forward sensitivity solution is available for postprocessing when the sensitivity solver has been used with the forward sensitivity method, and for the dependent variables that have been solved for. For scalar sensitivity variables, you can access the corresponding forward sensitivity solution by giving the sensitivity variable name as the second argument to this operator. For example, with the dependent variable u and the scalar sensitivity parameter q , you access the forward sensitivity solution $\partial u/\partial q$ as `sens(u, q)`. For a sensitivity parameter that is not scalar, a more elaborate syntax

specifying a unique degree of freedom must be used. This is done by giving an integer as the second argument, corresponding to the global degree of freedom number for the requested sensitivity parameter.

THE REALDOT OPERATOR

The expression `realdot(a,b)` treats complex numbers a and b as if they were real-valued vectors of length 2 and returns their dot product. You can also think of the operator call as a shorthand form of `real(a*conj(b))`. This expression, however, is not an analytical function of its complex arguments and therefore has no unique partial derivatives with respect to a and b . The difference between `realdot(a,b)` and `real(a*conj(b))` is that the partial derivatives of the former with respect to a and b are defined as `conj(b)` and `conj(a)`, respectively, while for the latter expression, the partial derivatives are `real(conj(a))` and `real(a)`.

The difference between the partial derivative definitions is important during sensitivity analysis of frequency-response problems (scalar or vector Helmholtz equations). Common objective function quantities like power and energy must be redefined in terms of `realdot(a,b)` rather than `real(a*conj(b))` for the sensitivity solver to compute correct derivatives. Note that this applies also to the absolute value, `abs(a)`, via the definition $|a|^2 = \text{realdot}(a,a)$.

Variable Naming Conventions

VARIABLE NAME SUFFIXES

Each application mode variable gets an application mode suffix to avoid naming conflicts. The suffix is the application mode name that appears in the Model Navigator and helps to clarify variable naming. If you change the application mode name, the variable name suffix changes accordingly.

For example, the variables derived from the application mode for a one-variable coefficient form PDE get the default suffix `_c` if this was the first application mode of this type. The next application modes of this type generates the suffixes `_c2`, `_c3`, and so on.

As another example, the default name for the Heat Transfer by Conduction application mode is `ht`. The name appears in the user interface when the application mode is active in your model. The dependent variable is the temperature T . To use the dependent variable in equations and postprocessing, type `T` in an edit field or in code. Likewise `Tx` is the temperature gradient in the x direction, and `Tt` is the temperature change in time

for time-dependent models. Expressions you input for coefficients and other application mode variables are designated by the suffix for the application mode name. By default the suffix is `_ht`. For example, to use the heat capacity in equations or postprocessing, type `C_ht` in an edit field.

For information on physics mode suffixes, see the chapter on physics modes in the *COMSOL Multiphysics Modeling Guide*.

NAMING OF SHAPE-FUNCTION VARIABLES AND EQUATION VARIABLES

The naming of shape-function variables and equation variables follows a certain convention. The index number denotes the equation number to which the expression corresponds. For variables not including any contribution from a dependent variable (the dependent variable is not part of the name), the equation index number appears immediately after the variable name, for example, `ga1x` or `g1`. The letters *x*, *y*, and *z* at the end of a variable name denote the spatial coordinate component. You can change the names of dependent variables (field variables) when adding an application mode and change the names of independent variables (spatial coordinates) when adding a geometry. This does not change the name of equation variables such as `au`, `beu`, and `dau`.

In the following descriptions of geometric variables and field variables, replace all letters in italic font with the actual names for the dependent variables (solution components) and independent variables (spatial coordinates) in the model. Replace *u* with the names of the dependent variables in the model, and replace *x*, *y*, and *z* with the first, second, and third spatial coordinate variable, respectively. x_i represents the *i*th spatial coordinate variable. If the model contains a deformed mesh, you can replace the symbols *x*, *y*, *z* with either the spatial coordinates (*x*, *y*, *z* by default) or the reference coordinates (*X*, *Y*, *Z* by default).

Geometric Variables

SPACE COORDINATE VARIABLES

The space coordinate variables (independent variables) *x*, *y*, *z* are available for all domain types. For a Cartesian geometry their default names are *x*, *y*, *z*. For axisymmetric geometries the space coordinates are normally called *r* and *z*. You can change the default names of the space coordinates when adding a geometry to a model (see “Changing the Names of the Space Coordinates” on page 29). If a deformed mesh is used, *x*, *y*, *z* can be both the *spatial coordinates* (*x*, *y*, *z* by default) and the

reference coordinates (X, Y, Z by default); see “Mathematical Description of the Mesh Movement” on page 454 of the *COMSOL Multiphysics Modeling Guide*.

If the model includes a deformed mesh, the variables $xTIME$, $yTIME$, $zTIME$ represent the mesh velocity. To access these variables, replace x , y , and z with the names of the spatial coordinates in the model (x , y , and z by default).

PARAMETERIZATION VARIABLES

The surface-boundary parameterization variables are useful for defining distributed loads and constraints such as a parabolic velocity profile. The available parameterization variables are:

- The curve parameter s in 2D. It is equivalent to the Bézier curve parameter, and it runs from 0 to 1 in the direction indicated by the arrows shown on the boundaries in boundary selection mode. You can use s on boundaries when specifying boundary conditions.
- The arc length parameters $s1$ and $s2$ in 3D are available on edges and are approximately equivalent to the arc length. In many cases they can be difficult to use because the relationship between x , y , and z (the space coordinates) and $s1$ and $s2$ is nonlinear. It is straightforward to use $s1$ and $s2$ in cases where a face is equivalent to an untrimmed patch, that is, a face that has no parts cut away. Often it is more convenient to use expressions with x , y , and z for specifying distributed boundary conditions. To see the values of $s1$ and $s2$, plot them using a surface or line plot.

Creating a Parabolic Velocity Profile

To create a parabolic velocity profile with a maximum velocity of v_0 on an inflow boundary in a 2D fluid-flow model, enter $v_0*4*s*(1-s)$ in the edit field for the appropriate velocity component (where v_0 is a constant that you define in the model).

The Chemical Engineering Module, MEMS Module, and Earth Science Module have application modes for fluid flow with a predefined option for a laminar inflow, which provides a parabolic velocity profile.

Showing and Hiding Direction Arrows

To control the display of the boundary-direction arrows, follow these steps:

- 1 From the **Options** menu choose **Preferences**.
- 2 In the **Preferences** dialog box click the **Visualization** tab.
- 3 In the **Show** area, select or clear the **Direction arrows** check box to show or remove the direction arrows.
- 4 Click **Apply**.

TANGENT AND NORMAL VARIABLES

The tangent and normal variables are components of the tangential and normal unit vectors.

Tangent Variables

- In 2D, t_x and t_y define the curve tangent vector associated with the direction of the boundary.
- In 3D, the tangent variables t_{1x} , t_{1y} , and t_{1z} are defined on edges. The tangent variables t_{1x} , t_{1y} , t_{1z} , t_{2x} , t_{2y} , and t_{2z} are defined on surfaces according to

$$(t_{ix}, t_{iy}, t_{iz}) = k_i \left(\frac{\partial x(s_1, s_2)}{\partial s_i}, \frac{\partial y(s_1, s_2)}{\partial s_i}, \frac{\partial z(s_1, s_2)}{\partial s_i} \right), \quad i = 1, 2$$

These most often define two orthogonal vectors on a surface, but the orthogonality can be ruined by scaling geometry objects. The vectors are normalized; k_i is a normalizing parameter in the expression just given.

If you use a deformed mesh, the tangent variables are available only for the undeformed mesh. In that case, replace x , y , and z with the names of the reference coordinates (X , Y , Z by default).

Normal Variables

- In 1D, n_x is the outward unit normal pointing out from the subdomain.
- In 2D, n_x and n_y define a normal vector pointing outwards relative to the subdomains.
- In 3D, n_x , n_y , and n_z define a normal vector pointing outwards relative to the subdomains.

Because the normal unit vector always points outward, it might be more convenient to use the vector components $-n_y$ and n_x to represent the tangential unit vector.

Direction of the Normal Component on Interior Boundaries

To get control of the direction of the normal component on interior boundaries, the following variables are available:

- In 1D:
 - un_x , the outward unit normal seen from the upper subdomain
 - dn_x , the outward unit normal seen from the lower subdomain

- In 2D:
 - unx and uny for the up direction
 - dnx and dny for the down direction

The upside is defined as the left side with respect to the parameterization of the curve in the variable s as described earlier in this chapter.

- In 3D:
 - unx , uny , and unz for the up direction
 - dnx , dny , dnz for the down direction

The up and down directions are with respect to the geometric parameterization of the boundary in the variables $s1$ and $s2$. The definition of the geometric normal direction is

$$n_{\text{geom}} = (t_{1x}, t_{1y}, t_{1z}) \times (t_{2x}, t_{2y}, t_{2z}).$$

The down-normal (defined by dnx , dny , and dnz) is equivalent to the geometric normal direction. The geometric normal is pointing into the upside of the geometry. The up-normal (defined by unx , uny , and unz) is pointing out from the upside of the geometry, for example, $dnx = -unx$, $dny = -uny$, and $dnz = -unz$.

To visualize any of these vector variables use arrow plots from the **Plot Parameters** dialog box.

If a deformed mesh is used, the normal variables are available both for the deformed configuration and for the undeformed configuration. In the first case, replace x , y , and z with the spatial coordinate names (x , y , and z by default). In the second case, replace x , y , and z with the reference coordinate names (X , Y , and Z by default).

Normal Vector Variables Representing Element Surface Normals

A similar set of variables— $n_{x\text{mesh}}$, $un_{x\text{mesh}}$, and $dn_{x\text{mesh}}$, where x is the name of a space coordinate—use the element shape function and are normal to the actual element surfaces rather than to the geometry surfaces.

Note: There are also outward normal component variables defined for each application mode. You can access these variables by appending an underscore plus the application mode's name, for example, n_{x_ps} for the unit normal x -direction component for a Plane Stress application mode. This is useful for interior boundaries

that are exterior boundaries for the application mode. For example, at the borders between a fluid domain and a solid domain, the outward normal component variables for the structural mechanics application mode (which is turned off in the fluid domain) always point outward from the solid domain into the fluid domain.

MISCELLANEOUS VARIABLES

- The variable `h`, available only in subdomains, represents the mesh element size (of the undeformed mesh), that is, the length of the longest edge of the element.
- The variable `dom` is the domain number—the subdomain number, the boundary number, the edge number, or the vertex number. This variable takes only integer values. For subdomains the variable `sd` has the same meaning.
- The variable `meshtype` is the mesh type index for the mesh element. This is the index into the cell array structure variable returned by the command `get(fem.mesh, 'el')`.
- The variable `meshelement` is the mesh element number. This is the index of the mesh element within a mesh type structure variable returned by the command `get(fem.mesh, 'el')`.
- The variable `dvol` is the *mesh element scale factor*, which is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates. For 3D domains, this is the factor that COMSOL Multiphysics multiplies volumes by when moving from local coordinates to global coordinates. In 2D and 1D domains, it is an area scaling factor and length scaling factor, respectively. If a deformed mesh is used, `dvol` is the mesh element scale factor for the undeformed mesh. The corresponding factor for the deformed mesh is named `dvol_app1`, where `app1` is the name of the application mode that controls the mesh deformation (`ale` or `pg`). To change the default name of the mesh element scale factor for the different frames, use the **Model Settings** dialog box on the **Physics** menu.
- The variable `qual` contains a mesh quality measure. See “Visualizing the Mesh Quality” on page 371 for more information.
- The variable `reldetjac` is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates, normalized to that of a first-order element. In other words, it is the determinant of the Jacobian matrix for the mapping from the straight mesh element to the possibly curved element used when solving. Use this variable to measure the difference in shape between a curved element and the corresponding straight element. The variable `reldetjacmin` is a

scalar for each element defined as the minimum value of the `re1detjac` variable for the corresponding element. A `re1detjacmin` value less than zero for an element means that the element is wrapped inside-out; that is, the element is an *inverted mesh element*. For more information on inverted mesh elements, see “Avoiding Inverted Mesh Elements” on page 374.

Using Field Variables

Field Variable Definition and Categories

The field variables are the dependent variables and variables derived from them.

SUBDIVISION OF FIELD VARIABLES

The field variables are divided into the following subcategories:

- Shape function variables—see “Shape Function Variables” on page 177.
- Application mode variables—see “Application Mode Variables” on page 180.
- Expression variables—“Expression Variables” on page 181.
- Coupling variables—see “Coupling Variables” on page 181.
- Equation variables (for PDE modes)—see “Equation Variables” on page 268 of the *COMSOL Multiphysics Modeling Guide*.
- Boundary-coupled equation variables (for PDE modes)—see “Boundary-Coupled Equation Variables” on page 270 of the *COMSOL Multiphysics Modeling Guide*.
- Special variables for the time, eigenvalues, and phasor—see “Special Variables” on page 181.

Shape function variables and boundary-coupled shape variables are related to the *shape functions* of the finite elements. The shape function variables are related to the degrees of freedom defined by a finite element.

Application mode variables are application-mode specific and consist of physically meaningful combinations of material properties and other variables. These variables are predefined for each application mode.

An expression variable is a user-defined shorthand for an expression involving other variables.

Coupling variables are user-defined variables for nonlocal couplings.

Equation variables and boundary-coupled equation variables are only accessible when you have selected the coefficient or general solution forms. These variables are shorthands for certain terms in the PDE and boundary conditions for the coefficient form or the general form. You normally use these variables only for postprocessing.

Special variables are variables that do not belong to any of the previous groups: variables that represent the time, eigenvalues, and phasor.

Shape Function Variables

A finite element defines a number of variables, typically a dependent variables and its derivatives. Such variables are called *shape functions variables* because they can be computed directly from shape functions and the degrees of freedom. When you select an application mode, COMSOL Multiphysics asks you to provide names for the dependent variables. COMSOL Multiphysics uses these variable names to construct the finite elements. Find the finite elements of an application mode by opening the **Element** page of the **Subdomain Settings** dialog box. For example, when using the Navier-Stokes application mode the **Shape functions** edit field looks like

```
shlag(2, 'u') shlag(2, 'v') shlag(1, 'p')
```

which means you have introduced two Lagrange finite elements of order two with dependent variable names *u* and *v*, respectively, and one Lagrange finite element of order one with dependent variable named *p*. The dependent variable name is the basis for additional variables names that the finite elements generate. For the Lagrange element the following table shows the available variable names, assuming you gave the name *u* as the argument to the shape function, and that you provided the names *x*, *y*, and *z* for the independent variables.

DOMAIN \ SPACE DIM	1D	2D	3D
POINT		u	u
EDGE			u, uTx, uTy, uTz
BOUNDARY	u, uTx, ut, uTxt	u, uTx, uTy, ut, uTxt, uTyt	u, uTx, uTy, uTz, ut, uTxt, uTyt, uTzt
SUBDOMAIN	u, ux, uxx, ut, uxt, uxxt, utt, uxtt, uxxtt,	u, ux, uy, uxx, uxy, uyx, uyy, ut, uxt, uyt, uxxt, uxyt, uyxxt, uyyt, utt, uxtt, uytt, uxxtt, uxytt, uyxxtt, uyytt	u, ux, uy, uz, uxx, uxy, uxz, uyx, uyy, uyz, uzx, uzy, uzz, ut, uxt, uyt, uzt, uxxt, uxyt, uxzt, uyxxt, uyyt, uyzt, uzxt, uzyt, uzzt, utt, uxtt, uytt, uzt, uxxtt, uxytt, uxztt, uyxxtt, uyytt, uyztt, uzxtt, uzytt, uzztt,

In the previous example with the Navier-Stokes application mode, you get the set of variables indicated above for *u*, *v*, and *p*, respectively.

The variables ux , uy , and uz are the components of the gradient ∇u , that is, the 1st-order spatial derivatives.

The variables uxx , uxy , uxz , uyx , uyy , uyz , uzx , uzy , and uzz are the 2nd-order space derivative components. They are meaningful only if the degree of the polynomial shape function used in an element is high enough. For elements with 2nd-degree polynomial shape functions (2nd-order elements), the polynomial degree of the 2nd-order derivatives is zero; that is, the second derivatives are constant in each element. For element orders lower than two, the second derivatives evaluate to zero regardless of the values of the 2nd-order derivatives of the true solution.

Other finite elements such as the vector element, `shvec`, define slightly different sets of variable names. See the section “Finite Elements” on page 482 of the *COMSOL Multiphysics Reference Guide* for lists of variable name defined by each finite element.

If the model uses a deformed mesh, each finite element is associated with a certain frame (the spatial frame or the reference frame) given by the **Frame** property in the **Shape functions** edit field. This frame is determined by the **Frame** property of the application mode that defines the finite element (see “Frames in the Model Navigator” on page 458 of the *COMSOL Multiphysics Modeling Guide*.) The frame determines the names of the variables generated by the finite element. For instance, if the spatial frame is used, the Lagrange element computes derivatives with respect to the spatial coordinates, ux , uy , uz . If the reference frame is used, the Lagrange element computes derivatives with respect to the reference coordinates uX , uY , uZ .

THE TIME DERIVATIVES OF THE DEPENDENT VARIABLES

The variable ut is the time derivative of the dependent variable u . You can also form mixed space-time derivatives as ux_t , for example, uxt ,

$$\frac{\partial^2 u}{\partial x \partial t}$$

Note that the t must appear last in a mixed derivative. You can also use second time derivatives such as utt or uxt (but not higher derivatives in time). To include the time derivatives in the solution output, you must select that option in the **Solver Manager** before you start a time-dependent analysis. See “Selecting Variables for Solution Output” on page 441 for details.

The variables with time derivatives evaluate to the correct value independently of the selected solution form. However, the Jacobian contributions for the time derivatives are accurate only when you have selected the weak solution form, which is the default.

If you change the solution form to coefficient form or general form, we recommend you use the d_a coefficient to introduce a time derivative into the problem.

If the model contains a deformed mesh, there is, in addition to the usual time derivative u_t , the *mesh time derivative* u_{TIME} (see “Mathematical Description of the Mesh Movement” on page 454 of the *COMSOL Multiphysics Modeling Guide*). This also holds for mixed space-time derivatives.

TANGENTIAL DERIVATIVE VARIABLES

On boundaries, edges, and points you also have access to *tangential derivative variables*. They have names such as u_{Tx} , u_{Ty} , and u_{Tz} . Using these variables, it is possible to create models with phenomena on boundaries, edges, or points as described with PDEs. The tangential derivative variables represent the Cartesian components of the tangential projection of the gradient of shape function variables:

$$(\nabla u)_T = (I - \mathbf{n}\mathbf{n}^T) \cdot \nabla u .$$

In this equation, $(\nabla u)_T$ is the tangential gradient, which consists of the tangential derivatives in each spatial direction, I is the unity tensor, \mathbf{n} is the outward unit normal vector, and ∇u is the gradient of u .

More information about the use of tangential derivative variables appears in the chapter “The Weak Form” in the *COMSOL Multiphysics Modeling Guide*.

The tangential derivative variables evaluate to the correct value independently of the selected solution form. However, the Jacobian contributions for the tangential derivative variables are accurate only when you have selected the weak solution form. Therefore avoid using another solution form than the weak form (which is the one that the automatic solution form selection chooses in almost all cases) when you use tangential derivative variables in an equation. In addition, to get a divergence operator acting on expressions involving the tangential derivative variables, you must use the weak form to enter the problem. See “Theoretical Background” on page 368 in the *COMSOL Multiphysics Modeling Guide* for an example.

For an example of the use of tangential derivative variables, see “Shell Diffusion” on page 156 of the *COMSOL Multiphysics Model Library*.

VARIABLE INHERITANCE

On boundaries, edges, and points, gradients and second derivatives of the shape functions are available by *inheritance*, that is, COMSOL Multiphysics computes the average of the values of the variables from the adjacent domains. This process can

progress for several levels. For example, ux is the average on a boundary from the adjacent subdomains, then the average on an edge from the adjacent boundaries, and finally, the average at the points from the adjacent edges. Avoid using variable inheritance for gradients and second derivatives in a model if possible. Instead, use the tangential derivative variables for PDE modeling on boundaries. For postprocessing of integrals of reaction forces and fluxes, use the `reacf` operator. Also, for high accuracy reaction forces and fluxes in other circumstances, use weak constraints on boundaries instead of directly accessing the gradient through inheritance (see “Computing Accurate Fluxes” on page 266 for more information).

BOUNDARY COUPLED SHAPE VARIABLES

Use the up and down operators to access values on either side of the boundary between subdomains (see “The up, down, and mean Operators” on page 165). Explicit variable names for the gradient of the dependent variables on either side of a boundary can be made available by selecting the **Boundary coupled shape variables** check box in the **Model Settings** dialog box.

Let u be the name of the dependent variable and x be the name of a space coordinate. Then the following variables are available.

TABLE 4-5: BOUNDARY COUPLED SHAPE VARIABLES

VARIABLE	MEANING
uux	The value of ux taken from the adjacent subdomain in the up direction (see page 172). If ux is not defined on that subdomain, the variable uux is undefined.
dux	The value of ux taken from the adjacent subdomain in the down direction (see page 172). If ux is not defined on that subdomain, the variable dux is undefined.

Application Mode Variables

Application mode variables are predefined variables that the application modes introduce. They are typically functions of the dependent variables. The *COMSOL Multiphysics Modeling Guide* provides documentation on these variables for each application mode. The application modes in the COMSOL add-on modules are documented in the respective module’s *User’s Guide*. To obtain a list of application mode variables, select the menu item **Physics>Equation System>Subdomain Settings**. Then click the **Variables** tab. Many application mode variables are also available in the **Predefined quantities** lists in the dialog boxes on the **Postprocessing** menu.

Application mode variables include a suffix that indicated the application mode. See “Variable Naming Conventions” on page 169 for more information.

Expression Variables

Expression variables are ones that you define to simplify the modeling. They are similar to application mode variables because they are functions of the dependent variables. To define expression variables, use the **Expressions** choice on the **Options** menu. You can define expression variables globally or on subdomains, boundaries, edges, and points. If you define an expression variable on boundaries, it also becomes available on adjacent edges and points by variable inheritance. You can use expression variables in a model and during postprocessing. See “Defining Constants, Expressions, and Coupling Variables” on page 139 for information on how to define expression variables.

Coupling Variables

The coupling variables can map the value of an expression of field variables by integration, extrusion, or projections. For documentation on the coupling variables, see “Using Coupling Variables” on page 269.

Special Variables

The special variables are those that do not have a natural place under the categories already presented. These include the time variable, the eigenvalue variable, and the phasor variable.

THE TIME VARIABLE

For time-dependent problems, the time variable t is available. It can be part of any expression in the point, edge, boundary, and subdomain settings, as well as in postprocessing. It is always a scalar, even for time-dependent problems where the solution contains more than one output time. The value of t during postprocessing corresponds to the selection made in the **Solution at time** list.

THE EIGENVALUE VARIABLE

When specifying an eigenvalue problem you can use the eigenvalue variable like any other variable. By default, the name of the eigenvalue variable is *lambda*, but you can change this in the **Solver Parameters** dialog box. The eigenvalue solver extracts the Jacobian, the damping matrix, and the mass matrix through Taylor expansion of the

problem with respect to the eigenvalue variable around a specified eigenvalue linearization point (which is zero by default). Other solvers treat the eigenvalue variable as a constant with value zero, unless it is set by an eigenvalue solution used as initial solution.

You can also specify eigenvalue problems like time-dependent problems. COMSOL Multiphysics then replaces the time derivative $\partial u/\partial t$ with $-\lambda u$, and the second time derivative $\partial^2 u/\partial t^2$ with $\lambda^2 u$, where λ is the eigenvalue. It also omits the source terms.

When specified using time derivatives, an eigenvalue problem in coefficient form reads

$$\nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + a u = \lambda d_a u - \lambda^2 e_a u$$

while on explicit form it is written as a static problem

$$\nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + a u = 0$$

where the coefficients may be functions of the eigenvalue variable. Clearly, it is possible to convert the first form to the second by adding $e_a \lambda^2 - d_a \lambda$ to the a coefficient in the latter. For more information on the eigenvalue equation see “Using the Eigenvalue Solver” on page 402.

Consider the common case where $e_a = 0$: a linear eigenvalue problem. For a 1-variable coefficient form eigenvalue PDE, the eigenvalue term is then

$$\lambda d_a u$$

where d_a is a scalar, and the eigenvalue λ is by default available as the eigenvalue variable (default name: `lambda`). When written as a time-dependent problem, the eigenvalue term factor $d_a u$ is available as the variable `dau`.

For a 2-variable coefficient form eigenvalue PDE, the first eigenvalue term is

$$\lambda d_a u = \lambda \begin{bmatrix} d_{a11} & d_{a12} \\ d_{a21} & d_{a22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \lambda \begin{bmatrix} d_{a11} u_1 + d_{a12} u_2 \\ d_{a21} u_1 + d_{a22} u_2 \end{bmatrix}.$$

The components of the vector on the right side are available as `dau1` and `dau2`.

After solving an eigenvalue problem, the eigenvalue name is available as a scalar variable for use in postprocessing expressions. To choose between different eigenvalues, select one from the **Eigenvalue** list in any dialog box in the **Postprocessing** menu. The value of the eigenvalue variable corresponds to the selection made in the **Eigenvalue** list.

THE PHASOR VARIABLE

In all physics application modes for the analysis of time-harmonic problems, COMSOL Multiphysics obtains the solution for a *phasor* variable, u , defined by

$$u(\mathbf{r}, t) = \text{Re}(\tilde{u}(\mathbf{r})e^{j\omega t}).$$

The phasor \tilde{u} contains the amplitude and phase information of u but is itself independent of t . The real part of \tilde{u} is the value of u at time $t = 0$. To obtain the value of u at any other time you can specify a phase angle, ϕ , in all postprocessing dialog boxes. This phase angle ϕ equals ωt_0 for a given time t_0 .

In the physics application modes, COMSOL Multiphysics multiplies the solution variable as well as all other time-harmonic application mode variables by the phase factor $e^{j\phi}$. When you define a problem using phasors with the PDE modes, the software multiplies only the solution variable by $e^{j\phi}$. For other harmonic fields, you must give the phase factor explicitly. Do so using the variable `phase`. Assume you have a quantity $F = F e^{j\omega t}$ denoting an explicit load, specified by setting the coefficient form source term `f` equal to F . To obtain the proper harmonic behavior of F , multiply F by `exp(j*phase)` during postprocessing.

The Phase Property and Animations

When making animations, you must slightly modify the interpretation of the `Phase` property of the postprocessing functions and the interpretation of the variable `phase`. In this case you are not interested in the value just at a single time but in the entire time evolution.

The *phase property* that you can specify in the postprocessing dialog boxes is the phase at time $t = 0$. Thus the time evolution is given by a factor $e^{j(\omega t + \phi)}$, where ϕ is taken from the phase property.

When specifying a phase factor `exp(j*phase)` using the variable `phase`, this factor is interpreted as the factor $e^{j(\omega t + \phi)}$. Thus `exp(j*phase)` is the factor $e^{j\phi}$ given by the phase property multiplied by the time evolution factor $e^{j\omega t}$.

Phasor Products

Be sure to take special care when evaluating products of phasors. Assume you want to evaluate the product uv of two time-harmonic variables u and v . The time evolution of uv is

$$u(t)v(t) = \text{Re}(\tilde{u}e^{j\omega t})\text{Re}(\tilde{v}e^{j\omega t}).$$

When evaluating uv at a given time or when animating the time evolution, be sure to give uv as $\text{real}(u) * \text{real}(v)$.

Using the relationship

$$\text{Re}(a)\text{Re}(b) = \frac{1}{2}\text{Re}(ab^* + ab)$$

where b^* is the complex conjugate of b , you can also express the time average of uv . Specifically,

$$\text{Re}(\tilde{u}e^{j\omega t})\text{Re}(\tilde{v}e^{j\omega t}) = \frac{1}{2}\text{Re}(\tilde{u}\tilde{v}^* + \tilde{u}\tilde{v}e^{2j\omega t}).$$

Because the time average of the second term vanishes, you can obtain the average by evaluating $0.5 * \text{real}(u * \text{conj}(v))$.

Specifying Model Settings

You can configure a number of settings independently for each geometry in a model. To specify these model settings, use the **Model Settings** dialog box, which you open by going to the **Physics** menu and choosing **Model Settings**.

The dialog box has multiple identical pages, one tabbed page for each geometry in the model.

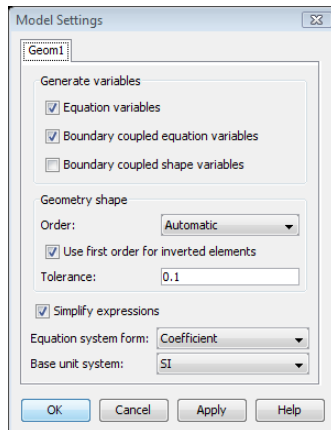


Figure 4-7: The Model Settings dialog box.

Generating Variables

In the **Generate variables** area you select which variables to generate for the selected geometry.

To generate some variables corresponding to certain equation terms, select the **Equation variables** check box. Examples of generated variables are:

- au, beu, dau, f, aLux, cux, and gax on subdomains
- g and qu on boundaries

To generate the values of some equation variables on the boundary in the up and down directions, select the **Boundary coupled equation variables** check box. Examples of generated variables are:

- u_{au} , u_{beu} , u_{cux} , u_{gax} , and similar. They represent the value of the shape variables on the boundary taken from the subdomain in the up direction.
- d_{au} , d_{beu} , d_{cux} , d_{gax} , and similar. They represent the value of the shape variables on the boundary taken from the subdomain in the down direction.

Note: To have access to the equation variables and the boundary-coupled equation variables, use the coefficient or general solution form.

For an overview of these variables, see “Special Variables” on page 181 in this book and “Boundary-Coupled Equation Variables” on page 270 of the *COMSOL Multiphysics Modeling Guide*.

To generate the values of the dependent-variable space derivatives on the boundaries in the up and down direction, select the **Boundary coupled shape variables** check box. Examples of generated variables are:

- u_{ux} , u_{uy} , and similar. They represent the value of the derivatives u_x and u_y on the boundary taken from the subdomain in the up direction.
- d_{ux} , d_{uy} , and similar. They represent the value of the derivative u_x and u_y on the boundary taken from the subdomain in the down direction.

Simplifying Expressions

Select the **Simplify expressions** check box to simplify the expressions generated for coefficients and variables. If COMSOL Multiphysics encounters a syntax error in an expression, it displays an error message.

Clearing this check box leaves the expressions as they are.

Selecting an Equation System Form

In the **Equation system form** list you can select among the equation system forms **Coefficient**, **General**, and **Weak**. This setting specifies in which form the **Equation System** dialog boxes present the system of equations. For more information about the three forms see “The PDE Modes” on page 246 and “Weak Form Modeling” on page 351

of the *COMSOL Multiphysics Modeling Guide*. The general equation system form differs from the form used in the PDE, General form application mode. The general equation system form uses the linearized form of the general form equations. See the section “The Linear or Linearized Model” on page 386.

Selecting a Geometry Shape Order

COMSOL Multiphysics computes the geometry shape order from the shape functions in the model. To set an explicit shape order, select one of the values from the **Geometry shape order** list. The geometry shape order is the order of the (Lagrange) shape functions used to describe the mesh.

Model Settings with Multiple Frames

When there is more than one frame in a geometry, the **Models Settings** dialog box makes it possible to change some settings for each frame.

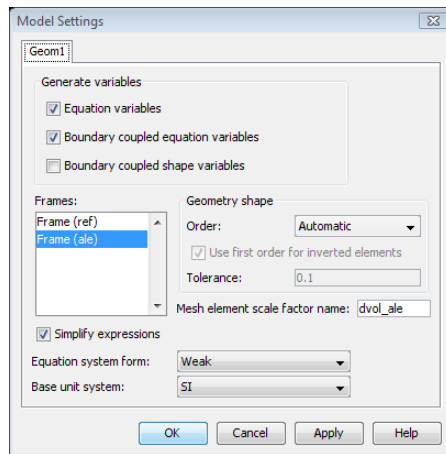


Figure 4-8: The Model Settings dialog box when there is more than one frame.

The **Geometry shape order** and **Mesh element scale factor name** settings apply to the frame selected to the left. The geometry shape order for all frames, except the reference frame, must be equal to the order of the shape functions defining the spatial coordinates (see “Subdomain Settings” on page 463 of the *COMSOL Multiphysics Modeling Guide*).

The mesh element scale factor is the scaling factor for the volume (area, length) for the mapping from local to global coordinates in a mesh element.

Using Units

Unit Systems in COMSOL Multiphysics

COMSOL Multiphysics supports the following unit systems:

METRIC UNIT SYSTEMS

- SI units, the International System of Units (SI, *Système International d'Unités*). This is the default unit system (sometimes also called MKS). For a list of SI units in COMSOL Multiphysics, see “SI Base Units, Derived Units, and Additional Units” on page 196.
- CGSA units. The CGS system uses centimeter, gram, and second as basic units of length, mass, and time, respectively. The remaining basic units are identical to the SI units. The CGS unit system gives nice values for small lengths, masses, forces, pressures, and energies when working on a microscale and with weak electromagnetic forces. The derived units of force, pressure, and energy have well-known and widely used names: dyne, barye, and erg, respectively. CGSA adds *ampere* as the basic unit for electric current. For a list of the CGSA units, see “Special CGSA Units” on page 200.
- Electromagnetic units (EMU). This system is based on Ampère’s law, which defines the unit of electric current once you select an appropriate value for the constant C . When dealing exclusively with magnetic effects, it is convenient to set $C = 1$. If CGS units are used for the remaining basic dimensions, the current unit is called an *abampere*, and the corresponding coherent unit system is called electromagnetic units. Unique names for derived units have been introduced by prefixing the SI name with *ab-*. For a list of EMU units, see “Special EMU Units” on page 201.
- Electrostatic units (ESU). Based on Coulomb’s law for the force between point charges, ESU uses a unit of charge called the *statcoulomb* with CGS units for length, mass, and time. From there, the *statampere*, or *franklin*, and other derived units of the electrostatic unit system follow. For a list of ESU units, see “Special ESU Units” on page 202.
- MPa units. For stationary structural mechanics, where the density does not appear in the equations, it can be convenient to use a system where the newton and megapascal (hence the name the MPa system) are naturally derived units of force and pressure. Keeping the SI unit for time, the basic units of length and mass

become millimeter and tonne. Except for the force and pressure units, other derived units are nameless. For a list of MPa units, see “Special MPa Units” on page 203.

ENGLISH UNIT SYSTEMS

- Foot-pound-second unit system (FPS units). The original foot-pound-second system seems to be the absolute system using the pound as a unit of mass. This version of the FPS system is in agreement with the IEEE standard (the pound is a unit of mass and not of force). The natural derived unit of force is the *poundal*. For a list of FPS units, see “Special FPS Units” on page 203.
- British engineering units. An alternative to the standard FPS system is the British engineering unit system (also called gravitational foot-pound-second system or foot-slug-second system). Here, the pound force is the natural unit of force, which causes the introduction of the mass unit *slug* such that a pound force is a slug-foot per second squared. For a list of British engineering units, see “Special British Engineering Units” on page 199.
- Inch-pound-second unit system (IPS units). It is possible to define varieties of the FPS and British engineering systems based on the inch instead of the foot as basic unit of length. This gives rise to two distinct inch-pound-second systems: the *absolute IPS system* (just called IPS) and the *gravitational IPS system*. For a list of IPS units, see “Special IPS Units” on page 203.
- Gravitational IPS units. This alternative IPS unit system considers the pound a unit of weight rather than a unit of mass. For a list of Gravitational IPS units, see “Special Gravitational IPS Units” on page 204.

OTHER

- Unitless (no units)

Specifying the Unit System

SETTING THE UNIT SYSTEM FOR THE CURRENT MODEL

To set the unit system for the current model:

- 1 From the **Physics** menu choose **Model Settings**.
- 2 In the **Model Settings** dialog box select the unit system in the **Base unit system** list.
- 3 Click **OK**.

You can also specify the unit system when adding a geometry in the Model Navigator:

- 1 Click the **Multiphysics** button.
- 2 Click the **Add Geometry** button.
- 3 In the **Add Geometry** dialog box select the unit for that geometry in the **Unit system** list.
- 4 Click **OK** to create a new geometry with the selected unit system.

This means that it is possible to have different unit systems in each geometry.

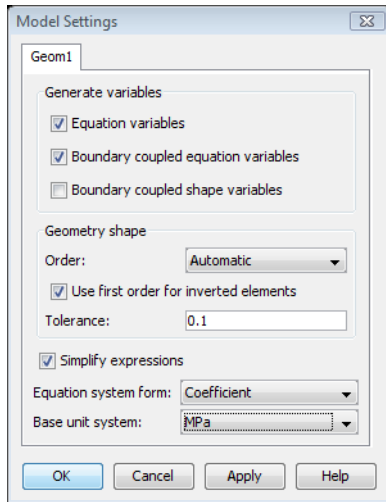


Figure 4-9: The Model Settings dialog box with MPa units as the base unit system.

CHOOSING THE DEFAULT UNIT SYSTEM AND UNIT DISPLAY

Using the **Preferences** dialog box, you can specify the default unit system and also if the user interface displays units or not:

- 1 From the **Options** menu choose **Preferences**.
- 2 Click the **Modeling** tab.
- 3 In the **Units** area select the default unit system from the **Default unit system** list.
- 4 Select or clear the **Show units** check box to show or hide the display of units in the user interface.
- 5 Clear the **Show expected units** check box to hide the unit column in the dialog boxes for the physics settings.

6 Select the **Highlight unexpected units** check box to make the unit appear red in the dialog boxes for the physics settings if the unit of an entered expression is invalid or differs from the expected unit.

7 Click **OK**.

The highlighting of unexpected units also includes displaying a tooltip in the edit field that contains a value or expression with an unexpected unit. The tooltip displays **Inconsistent unit** if the expression in the edit field does not contain a unit. If the expression contains a unit, the tooltip displays the unexpected base unit. For example, if you type `3 [ft]` into an edit field for a density in a model using SI units, the tooltip displays **Unexpected base unit: [m]**.

You can also select the default unit system for a model directly in the Model Navigator:

1 Click the **Settings** tab.

2 Select the unit system in the **Unit system** list.

3 Click **OK** when your Model Navigator entries are complete.

Changes to the unit display take effect immediately. Changing the default unit system affects all new models from then on (until you change the default setting again). It also affects the unit system setting for the current geometry of the current model.

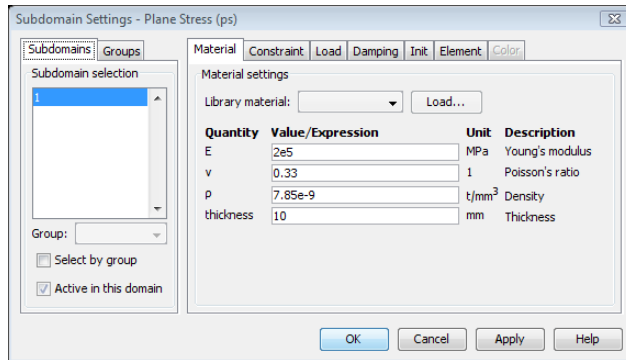


Figure 4-10: The Subdomain Settings dialog box for a Plane Stress application mode using MPa units.

In the graphical user interface, units appear in all physics settings dialog boxes and in the postprocessing dialog boxes. For postprocessing, you always have access to the corresponding units from all unit systems. Select one of the suggested units from the list box or type in your own unit that is compatible with the dimension of the postprocessing quantity.

For basic SI units, you can choose to scale data using the standard prefixes for powers of 10 such as *kilo*, *mega*, *milli*, *micro* according to the following table (you can use either the full prefix or the symbol but you must use the same form for the unit; that is, [milliampere] and [mA] are valid but not [mampere] or [milliA]):

TABLE 4-6: SI PREFIXES

FULL PREFIX	SYMBOL	FACTOR
yotta	Y	10^{24}
zetta	Z	10^{21}
exa	E	10^{18}
peta	P	10^{15}
tera	T	10^{12}
giga	G	10^9
mega	M	10^6
kilo	k	10^3
hekto	h	10^2
deca	dk	10^1
deci	d	10^{-1}
centi	c	10^{-2}
milli	m	10^{-3}
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}
atto	a	10^{-18}
zepto	z	10^{-21}
yocto	y	10^{-24}

Using the Unit Syntax

You can use the unit syntax to specify a quantity with any applicable unit regardless of the chosen base unit system. To do so, append the unit to any constant or variable in a model using a syntax where you enclose the unit in brackets, for example, `200[ft]` and `3e6[kg/m^3]`. You can use both the name and the symbol for a unit. For example, `2.4[ampere]` and `2.4[A]` are both valid to indicate an electric current in SI units (see “SI Base Units, Derived Units, and Additional Units” on page 196). The SI units can

also contain standard prefixes. Note that appending a unit means that you *multiply* the constant or variable to the left of the unit declaration with this unit. This multiplication takes precedence over other operators so, for example, $1/2[\text{m}]$ evaluates to 0.5 m^{-1} ($0.5[1/\text{m}]$) whereas both $(1/2)[\text{m}]$ and $1/2*1[\text{m}]$ evaluate to 50 cm ($0.5[\text{m}]$). Also, if L is a variable defined as $2[\text{m}]$, $L[1/\text{s}]$ evaluates to $2[\text{m}/\text{s}]$. The following examples show how you can use the unit syntax:

- Adding two quantities of the same kind that use different units: $0.5[\text{ft}]+33[\text{mm}]$. COMSOL Multiphysics converts the result to the base length unit for the model.
- Using multiplication with a unit to get consistent units for two quantities that you want to add, for example, $14[\text{kg}]+\rho_{\text{ht}}[\text{m}^3]$, which works if ρ_{ht} represents the density for a model using SI units.
- Making functions work independently of the unit system for a model. For example, assume you have a function $k(T)$ that defines the thermal conductivity k as a function of the temperature T and returns the conductivity in $\text{W}/(\text{m}\cdot\text{K})$ based on the temperature in kelvin. You can make this function also work with a unit system that uses degrees Fahrenheit for temperature (and a different unit for the thermal conductivity). To do so, use the following syntax to call the function:

```
k(T[1/K])[W/(m*K)]
```

When you use degrees Fahrenheit as the temperature unit, multiplying by $[\text{K}]$ provides a conversion from kelvin to degrees Fahrenheit. Conversely, multiplying by $[1/\text{K}]$ provides a conversion from degrees Fahrenheit to kelvin so that $T[1/\text{K}]$ converts the value of the temperature T from the base unit system to kelvin. This means that the output from the function $k(T)$ always has the same numerical value regardless of the unit system that you use. The multiplication with $[1/\text{K}]$ also makes the input dimensionless. The output from a function is also dimensionless.

Multiplying the thermal conductivity with $[\text{W}/(\text{m}\cdot\text{K})]$ provides a unit for k and a conversion to the base unit system's unit for thermal conductivity.

It is possible to add constants (without units) to any quantity. COMSOL Multiphysics then assumes that this value has the unit of the base unit system for the model. You can also concatenate several units, for example, $3.6[\text{N}][\text{m}]$, which is equivalent to $3.6[\text{N}\cdot\text{m}]$ and evaluates to $3.6[\text{J}]$.

The material libraries in COMSOL Multiphysics employ this unit syntax to make it possible to use the SI-based materials data with any base unit system.

DECLARING UNITS FOR CONSTANTS AND EXPRESSION VARIABLES

If you use constants or expression variables in the physics settings dialog boxes, it is good practice to define them using the unit syntax. The **Constants** dialog box and the dialog boxes for expression variables display the resulting unit, in the base unit system, of the constants and expressions that you define. You can then verify that these variables have the expected unit before using them in the physics settings. The unit of constants and expression variables are otherwise undefined. Using properties with undefined units in a model does not affect the numerical results from the analysis, but in the postprocessing stages, expressions involving such constants and variables are also unitless. Using other units than the standard units for the chosen base unit system, or using SI prefixes, the conversion to the base unit system's unit also affects the value (quantity) using a scaling factor (and an offset in the case of temperature units). The **Value** column in the **Constants** dialog box displays the quantity and unit in the base unit system so that you can see the result of the unit conversion. For example, if you define a constant as $3[\text{ft}]$ for a model using SI units, the result in the **Value** column is $0.9144[\text{m}]$.

INDICATION OF UNEXPECTED UNITS

If you select the **Highlight unexpected units** check box on **Modeling** page in the **Preferences** dialog box, the unit display appears red for the properties in the physics dialog boxes that have an invalid unit or a different unit than the expected one that the dialog box shows. A tooltip displays an error message at the corresponding text field. In the case of a valid but unexpected unit, this message contains the unexpected unit in the model's base unit system.

Switching Unit System

If you switch the unit system during modeling or use a unit system other than SI, COMSOL Multiphysics does not convert the data in the model with the following exceptions:

- COMSOL Multiphysics converts the default values for all application scalar variables to the current unit system when you create an application mode. Switching the unit system resets the values of the application scalar variables to their default values and converts them to the current unit system.
- The default values for all physics settings are in the current unit system (COMSOL Multiphysics converts them from the original values in SI units both when starting an application mode and when switching unit system). The values for any data in the current model do not change.

All data in the materials libraries are in SI units, but you can use them with any unit system because COMSOL Multiphysics converts these values to the corresponding values in the model's unit system. Note, however, that you must also declare the units of other input data for the model using the unit syntax (or convert the numerical values to the new units system). If the length unit changes, you must also convert the geometry dimensions to keep the size of the geometry using another length unit.

About Temperature Units

The relations between different temperature units involve an offset in addition to the usual scale factor. In many cases, the offset is not important to the physics equations because these equations are concerned only with temperature differences. There are, however, some cases where you must use an absolute or thermodynamic temperature measure. The most common example is the Stefan-Boltzmann law for black-body radiation used in radiation boundary conditions.

All predefined metric unit systems use the kelvin, which is an absolute temperature, as the basic unit of temperature. The English unit systems use the degree Fahrenheit as the basic unit of temperature, which, because the Fahrenheit scale is not absolute, is fine for most purposes except radiation. When modeling radiation in English units, you have to treat the temperature values as being given in degrees Rankine (which is the corresponding absolute scale). By doing so, the radiation terms are correct, but conversion to other units in postprocessing fails.

To declare temperature units in kelvin, use [K]. For degrees Fahrenheit use [degF], and for degrees Celsius use [degC].

DIFFERENTIAL VS. ABSOLUTE TEMPERATURE

If the dimension of an expression that includes a unit is temperature or 1/temperature, COMSOL Multiphysics interprets the dimension as an absolute temperature. If the dimension is something other than temperature but the unit expression includes temperature, the temperature is a differential temperature; that is, COMSOL Multiphysics uses no offset when converting between different temperature units.

The following examples illustrate how the unit conversion works for different expressions that include temperature units:

- `100[degC]` is an expression that has temperature as the dimension. COMSOL Multiphysics interprets it as an absolute temperature and evaluates it as 373.15 K.

- $373.15[1/K]$ is also interpreted as an absolute temperature (but no conversion is necessary from kelvin to kelvin).
- $100[\text{degC}/K]$ is dimensionless, and the temperature is therefore a differential temperature; that is, the result is 100 because the conversion uses no offset.
- To make COMSOL Multiphysics interpret the previous example as an absolute temperature, split the expression using two separate expressions such as $100[\text{degC}] * 1[1/K]$, which equals 373.15. This is also what occurs when you use a variable, for example, TC, defined as $100[\text{degC}]$. $TC[1/K]$ is then also two expressions where both are interpreted as absolute temperature.

SI Base Units, Derived Units, and Additional Units

The SI units form an internationally accepted system with seven units for base quantities and a large number of derived units.

Table 4-7 lists the SI units for the seven base quantities plus the radius.

TABLE 4-7: BASE SI UNITS

BASE QUANTITY	NAME	SYMBOL
length	meter, metre	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd
radius	meterradius, metrerradius	m, mr

COMSOL Multiphysics supports the following SI derived units:

TABLE 4-8: SI DERIVED UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
absorbed dose	gray	Gy
capacitance	farad	F
conductance	siemens	S
dose equivalent	sievert	Sv
electric charge	coulomb	C
electric resistance, impedance, reactance	ohm	Ω

TABLE 4-8: SI DERIVED UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
electric potential difference, voltage	volt	V
energy, work, heat	joule	J
force, weight	newton	N
frequency	hertz	Hz
illuminance	lux	lx
inductance	henry	H
luminous flux	lumen	lm
magnetic flux	weber	W
magnetic flux density, magnetic induction	tesla	T
plane angle	radian	rad
power	watt	W
pressure	pascal	Pa
radioactivity	becquerel	Bq
solid angle	steradian	sr

Note: To declare the SI unit for electric resistance, ohm, use [ohm]. COMSOL Multiphysics then displays this as [Ω].

You can also use the following additional units:

TABLE 4-9: ADDITIONAL UNITS IN COMSOL MULTIPHYSICS

DERIVED QUANTITY	NAME	SYMBOLS
energy	calorie	cal
energy	electronvolt	eV
length	mile	mi
length	nautimile	nmi
mass	ton, tonne	t
mass	gram	g
plane angle	degree	deg
pressure	atmosphere	atm
pressure	bar	bar
pressure	torr	Torr, mmHg
speed	mph, MPH	mph
speed	knot	knot
temperature	celcius	degC
time	year	a
time	day	d
time	hour	h
time	minute	min
volume	liter, litre	L, l

- The *calorie* is a small calorie or gram calorie, which equals 4.184 J. A large calorie or kilogram calorie is 1000 calories (4.184 kJ). Use [kca1] for large calories.
- The *mile* is the international statute mile, which equals 1609.344 m.
- The *nautical mile* (nautimile) is the nautical mile, which equals 1852 m.
- The *knot* is the same as nautical miles per hour.
- The *year* is a Gregorian year, which equals 365.2425 days.

The following table includes some other SI derived units that do not have special names and symbols:

TABLE 4-10: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

QUANTITY	NAME	SYMBOL
acceleration	meter per second squared	m/s ²
amount-of-substance concentration	mole per cubic meter	mol/m ³

TABLE 4-10: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

QUANTITY	NAME	SYMBOL
area	square meter	m ²
current density	ampere per square meter	A/m ²
heat capacity, specific heat	joule per kilogram kelvin	J/(kg·K)
magnetic field strength	ampere per meter	A/m
mass density	kilogram per cubic meter	kg/m ³
permeability	henry per meter	H/m
speed, velocity	meter per second	m/s
wave number	reciprocal meter	m ⁻¹
volume	cubic meter	m ³

You can use the symbols for these and other units when declaring units in COMSOL Multiphysics (for example, 10[m/s] for a velocity).

Special British Engineering Units

The base units in the British engineering unit system are identical to the SI units with the following exceptions:

TABLE 4-11: SPECIAL BASE UNITS IN THE BRITISH ENGINEERING UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	slug	N/A
temperature	Fahrenheit	degF
radius	footradius	ft, ftr

There is one derived unit that differs from the corresponding SI unit:

TABLE 4-12: DERIVED BRITISH ENGINEERING UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	pound-force	lbf

There is an additional unit for energy, Btu:

TABLE 4-13: ADDITIONAL BRITISH ENGINEERING UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
energy	British thermal unit	Btu

Refer to the British thermal unit using the symbol only, for example, $0.28[\text{Btu}/(\text{h}\cdot\text{in}\cdot\text{degF})]$ for a thermal conductivity.

Note: If you specify the British engineering unit system as the base unit system, COMSOL Multiphysics constructs derived units from the applicable SI base units and the units listed in Table 4-11 and Table 4-12. This means, for example, that the unit for voltage displayed in the user interface is lbf-ft/As rather than V (volt). In an edit field that expects a voltage as input, you therefore need to append a unit bracket when entering a numerical value, for example, 10[V].

Special CGSA Units

The base units in the CGSA unit system are identical to the SI units with the following exceptions:

TABLE 4-14: SPECIAL BASE UNITS IN THE CGSA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
radius	centimeterradius	cm, cmr

The CGSA unit system includes the following derived units that differ from the corresponding SI units:

TABLE 4-15: DERIVED CGSA UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	gal	N/A
energy	erg	N/A
force	dyne	N/A
pressure	barye	N/A
speed	kyne	N/A

Special EMU Units

The base units in the EMU unit system are identical to the SI units with the following exceptions:

TABLE 4-16: SPECIAL BASE UNITS IN THE EMU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
electric current	abampere, biot	N/A
radius	centimeterradius	cm, cmr

The EMU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 4-17: DERIVED EMU UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	gal	N/A
capacitance	abfarad	N/A
conductance	absiemens	N/A
electric charge	abcoulomb	N/A
electric resistance	abohm	N/A
electric potential difference, voltage	abvolt	N/A
energy	erg	N/A
force	dyne	N/A
inductance	abhenry	N/A
magnetic flux	abweber	
magnetic flux density	abtesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special ESU Units

The base units in the ESU unit system are identical to the SI units with the following exceptions:

TABLE 4-18: SPECIAL BASE UNITS IN THE ESU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
electric current	statampere, franklin	N/A
radius	centimeterradius	cm, cmr

The ESU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 4-19: DERIVED ESU UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	gal	N/A
capacitance	statfarad	N/A
conductance	statsiemens	N/A
electric charge	statcoulomb	N/A
electric resistance	statohm	N/A
electric potential difference, voltage	statvolt	N/A
energy	erg	N/A
force	dyne	N/A
inductance	stathenry	N/A
magnetic flux	statweber	N/A
magnetic flux density	stattesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special FPS Units

The base units in the FPS unit system are identical to the SI units with the following exceptions:

TABLE 4-20: SPECIAL BASE UNITS IN THE FPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	pound	lb
temperature	Fahrenheit	degF
radius	footradius	ft, ftr

There is one derived unit that differs from the corresponding SI unit:

TABLE 4-21: DERIVED FPS UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	poundal	N/A

Special IPS Units

The base units in the IPS unit system are identical to the SI units with the following exceptions:

TABLE 4-22: SPECIAL BASE UNITS IN THE IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	pound	lb
temperature	Fahrenheit	degF
radius	inchradius	in, inr

Special MPa Units

The base units in the MPa unit system are identical to the SI units with the following exceptions:

TABLE 4-23: SPECIAL BASE UNITS IN THE MPA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	millimeter	mm
mass	tonne, ton	t
radius	millimeterradius	mm, mmr

There is one derived unit that differs from the corresponding SI unit:

TABLE 4-24: DERIVED MPA UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
pressure	megapascal	MPa

Special Gravitational IPS Units

The base units in the Gravitational IPS unit system are identical to the SI units with the following exceptions:

TABLE 4-25: SPECIAL BASE UNITS IN THE GRAVITATIONAL IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	GIPS_mass	N/A
temperature	Fahrenheit	degF
radius	inchradius	in, inr

The following derived units differ from the corresponding SI units:

TABLE 4-26: DERIVED GRAVITATIONAL IPS UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	pound-force	lbf
pressure	psi	psi

Specifying Physics Settings

From the **Physics** menu you can specify all the physics and equations that define a model including:

- Equations
- Boundary and interface conditions
- Material properties
- Initial conditions

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

Specifically, the settings are available on the following physical domains:

- Subdomains
- Boundaries
- Edges
- Points
- Scalar variables that are independent of the geometry

Not all of these options are available for all geometry types and application modes.

All dialog boxes for the specification of the physics and equations accept variables, constants, and expressions as input data (see “Variables and Expressions” on page 138).

Opening the Physics Settings Dialog Boxes

To open all the physics settings dialog boxes:

- Go to the **Physics** menu and choose the corresponding command, for example **Boundary Settings** to open the **Boundary Settings** dialog box.
- In the domain selection modes, double-click a domain to select it and open the associated dialog box. For example, double-clicking a boundary opens the **Boundary Settings** dialog box and shows the present settings for the selected domains. For more on selection methods see “General Object Selection Methods” on page 126.
- In the **Model Tree**, using the Overview setting, right-click on the application mode for which you want to specify physics settings and select the appropriate dialog box.

In the Detail view and the Inspect view, click the plus sign next to the application mode of interest to open that part of the tree structure. Then right-click on the settings that you want to specify, for example, **Subdomain Settings**.

- Shift-click on a domain-mode selection button in the Main toolbar to open the associated dialog box. For example, shift-click the **Boundary Mode** button to open the **Boundary Settings** dialog box.

There are also several keyboard shortcuts:

- Press F5 to open the **Point Settings** dialog box (in 2D and 3D)
- Press F6 to open the **Edge Settings** dialog box (in 3D only)
- Press F7 to open the **Boundary Settings** dialog box
- Press F8 to open the **Subdomain Settings** dialog box

Working with Domain Groups

In the domain settings dialog boxes, you can work with individual domains or with groups of domains. All domains in a group have identical settings.

In a domain settings dialog box, the **Boundary Settings** dialog box, for example, click the **Group** tab to work with domain groups.

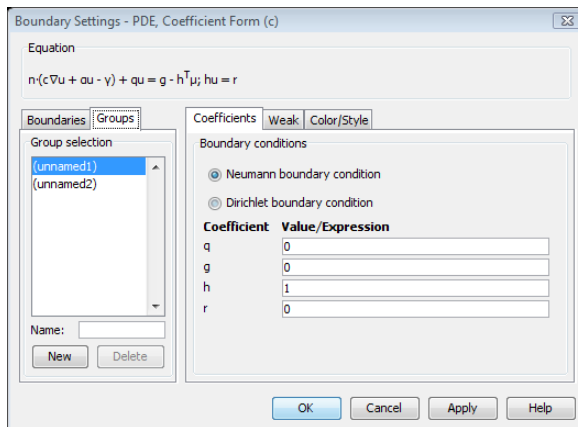


Figure 4-11: The group settings in the Boundary Settings dialog box.

When you choose a domain group, COMSOL Multiphysics selects the corresponding domains. Also, when you click the **Group** tab, COMSOL Multiphysics updates the domain selection to contain entire domain groups.

You can change the name of the group in the **Name** edit field. The default groups and the groups that you created due to changes made on subsets of other domain groups are initially unnamed.

Group names appear together with the domain label in the drawing area if you select the **Group names** check box on the **Visualization** page in the **Preferences** dialog box. To display the group names, you must also select the corresponding check box for the domain labels.

Create new groups by clicking the **New** button. They have no domains assigned to them, so you can create groups at any time for later use in the model. If you give a group the name Default, its settings becomes the default for every new domain and overrides the defaults for the application mode.

Groups are unused if they have no domains assigned to them. Delete unused groups by clicking the **Delete** button. You cannot delete used groups.

COMSOL Multiphysics automatically deletes unnamed groups if they are no longer used. Named groups, however, remain even if you, for example, delete the geometry.

To assign domains to a group, go to, for example, the **Boundary** page. Select a set of domains and choose a group name from the **Group** list under the **Boundary selection** list. The selected group name now appears to the right of the selected domain labels in the list. Note that you can only assign named groups to the selected domains.

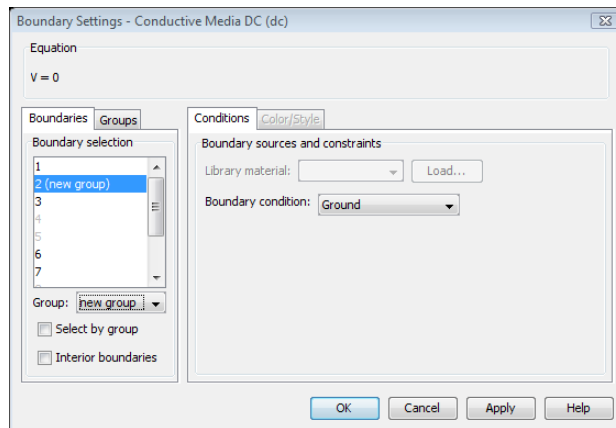


Figure 4-12: Group indication in the Boundary selection list.

In most application modes, there are different sets of boundary-condition types available for boundaries and interior boundaries. To be able to select among these sets

of boundary-condition types for unused groups there is a **Type** list. This list is unavailable for used domain groups.

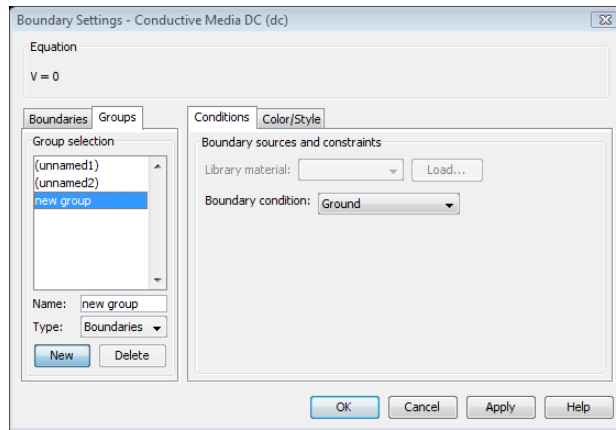


Figure 4-13: Selecting the boundary-condition types in the Type list.

At least for boundaries, the domains belonging to the groups automatically get different colors and line styles depending on the boundary-condition type in the selected application mode. You can override these settings with manual settings using the **Color** tab in 1D and 3D and the **Color/Style** tab in 2D.

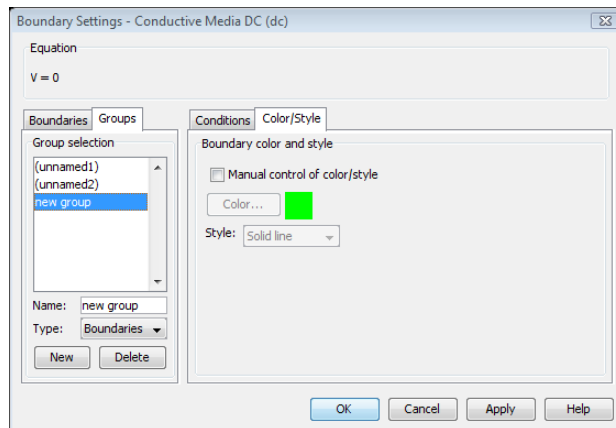


Figure 4-14: The Color/Style page with settings for boundary colors and line styles.

To specify a color for a subdomain in a 3D model, for example, open the **Subdomain Settings** dialog box, click the **Color** tab, and select the **Manual control of color** check box.

Then click the **Color** button to select a color. Add additional groups to display different colors in different parts of the geometry. The default setting is not to render faces in 3D, so click the **Render Face** button in the Visualization/Selection toolbar. Also notice that the **Highlight Face** button is selected by default. In that case, display of the domain group's color applies only to the domains that are not selected.

To switch off the group color and line style, clear the **Group style** check box on the **Visualization** page in the **Preferences** dialog box

The predefined multiphysics couplings use groups to predefine couplings between the fields of physics in the participating application modes. For more information about predefined multiphysics couplings see “Multiphysics Modeling” on page 375 of the *COMSOL Multiphysics Modeling Guide*.

Specifying Subdomain Settings and PDE Coefficients

The subdomain settings describe the physics on a model's main domain, which is divided into subdomains. You can set different values for each subdomain using settings of the following types:

- Coefficients that define the PDE on the subdomain. The PDE coefficient are available for PDE modes and the weak form modes.
- Material properties, sources, and sinks that define the physics in the subdomain. These properties are available in the physics modes and in the COMSOL Multiphysics modules. The software performs the translation into PDE coefficients automatically.

You can specify PDE coefficients for any COMSOL Multiphysics model using the **Subdomain Settings** dialog box for the equation system. See “Viewing and Modifying the Full Equation System” on page 218 for more information about the equation system view.

Using the Subdomain Settings Dialog Box

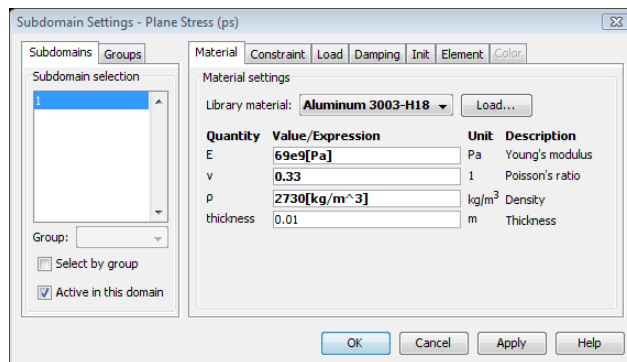


Figure 4-15: The Subdomain Settings dialog box for the Plane Stress structural mechanics application mode.

- 1 From the **Physics** menu, choose **Subdomain Settings**.

- 2 In the **Subdomain Settings** dialog box, make entries for the selected application mode and analysis type.

For a thorough description of the different application modes see the *COMSOL Multiphysics Modeling Guide*. The equations that the application mode solves appear at the top of the dialog box.

If you select multiple subdomains, the PDE coefficients and other settings you specify apply to all the selected subdomains. If the settings on the currently selected subdomains differ, the edit fields for those subdomains are locked, which they indicate with a yellow background. To clear the lock and overwrite the current settings for the selected domains, enter a new value or simply put the cursor in the edit field and press Enter.

DISPLAYING THE EQUATIONS

The **Subdomain Settings** dialog box for most application modes provide a display of the PDE that the application mode creates. The default setting is to show the equation. To show or hide the equations, do the following steps:

- 1 From the **Options** menu, choose **Preferences**.
- 2 In the **Preferences** dialog box, click the **Modeling** tab.
- 3 Click to select or clear the **Show equations** check box to show or hide the equations.
- 4 Click **OK**.

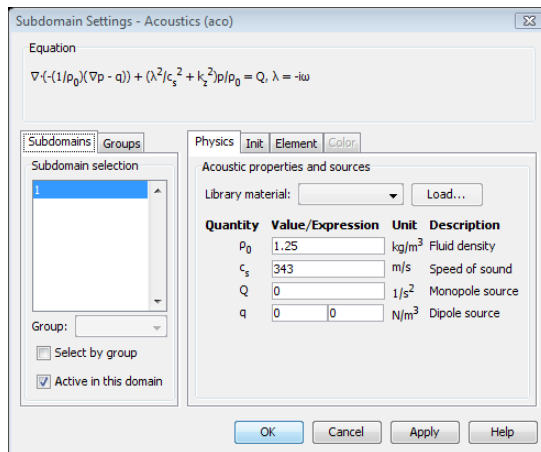


Figure 4-16: The Subdomain Settings dialog box for the Acoustics application mode, displaying the equation.

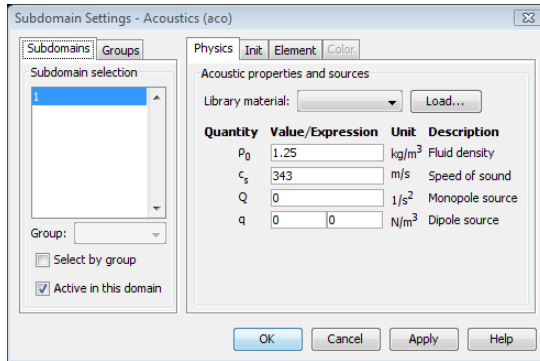


Figure 4-17: The Subdomain Settings dialog box for the Acoustics application mode, hiding the equation.

This setting also applies to the equations in the **Boundary Settings** dialog box, and in the corresponding dialog boxes for the equation system.

Specifying PDE Coefficients and Material Properties

To specify PDE coefficients or material properties click the **Coefficients** tab. You can define each of the coefficients as an expression using numeric values, variables, and constants as well as mathematical and logical function. For an overview of the variables available for use in expressions see “Variables and Expressions” on page 138.

SUBDOMAIN SETTINGS FOR COEFFICIENT FORM PDE COEFFICIENTS

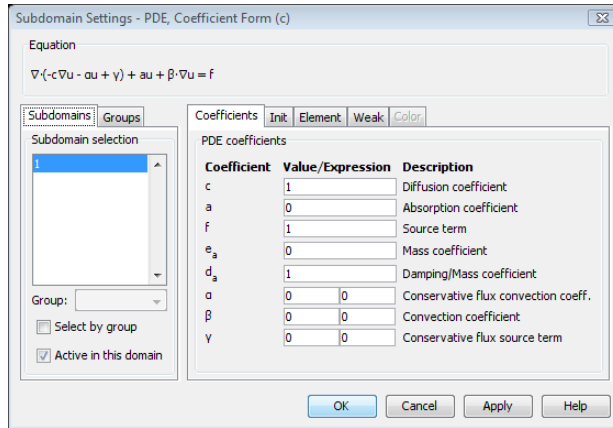


Figure 4-18: The Subdomain Settings dialog box for a scalar PDE, Coefficient Form application mode.

Figure 4-18 shows the **Subdomain Settings** dialog box for standard PDE coefficients in a 2D model. The PDE coefficient formulation appears at the top of the dialog box.

On the **Coefficients** page, enter expressions for the coefficients c , a , α , β , γ , f , d_a , and e_a that define the PDE in coefficient form.

The α , β , and γ coefficients are vectors of the same size as the space dimension. Enter the vector components in individual edit fields. The c coefficient can be a tensor that represents anisotropic material properties (see “Modeling Anisotropic Materials” on page 221).

Figure 4-19 depicts the **Subdomain Settings** dialog box for the coefficient system for the case with two dependent variables, u_1 and u_2 :

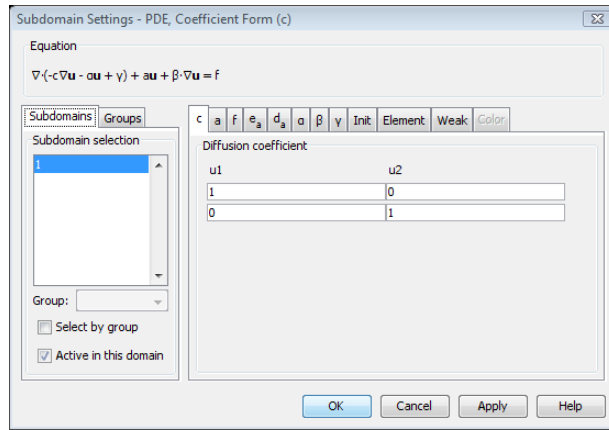


Figure 4-19: The diffusion coefficient (c) specification in the Subdomain Settings dialog box for a PDE, Coefficient Form model with two dependent variables, u_1 and u_2 .

The **Subdomain Settings** dialog box for PDE models with more than one dependent variable is a tabbed dialog box with one page for each PDE coefficient.

DEACTIVATING PHYSICS OR EQUATIONS IN SUBDOMAINS

You can deactivate the dependent variables for any of the application modes in the model in one or more subdomains by clearing the **Active in this domain** property. This way a model can include a physics that takes part in a subset of the total geometry.

Specifying Initial Conditions

In some types of analyses you must provide initial conditions:

- As the initial condition for a time-dependent analysis

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

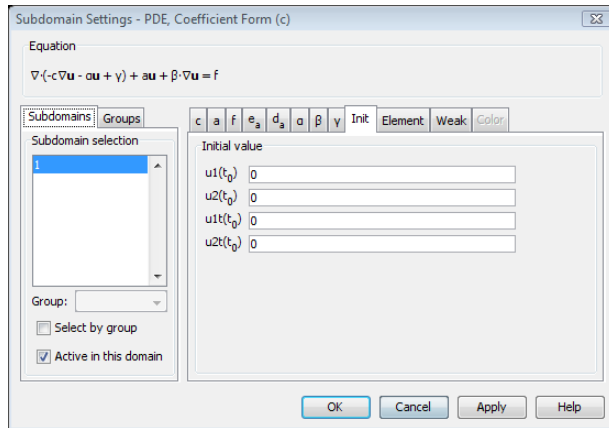


Figure 4-20: The initial condition specification page in the Subdomain Settings dialog box.

To specify initial conditions, click the **Init** tab.

Just as with coefficients, you can specify initial conditions on selected subdomains. Enter an initial value for each dependent variable. The default initial value is zero.

For some application modes you can also enter initial values for the first time derivative of the dependent variables. These are used when solving time-dependent problems containing second time derivatives.

For more information about using initial conditions and restarting the solver using a previous solution as an initial value see “The Solver Manager and Solver Scripting” on page 436.

Specifying the Finite Element Type

In addition to specifying the finite element type for a model in the **Model Navigator**, you can control the element settings from the **Subdomain Settings** dialog box.

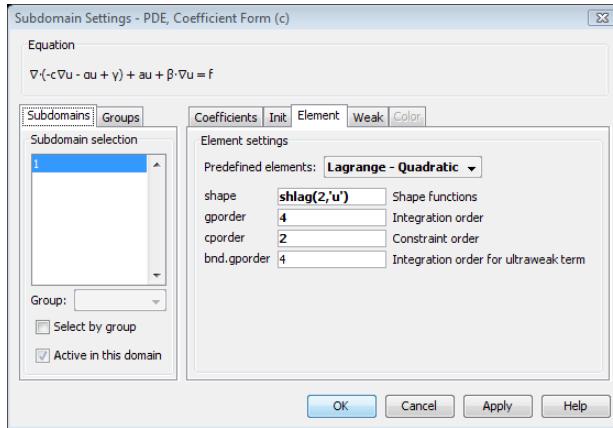


Figure 4-21: Specification of finite element settings in the Subdomain Settings dialog box.

Click the **Element** tab to specify the finite element settings. For the selected subdomains you can specify:

- The element shape function
- The integration order
- The constraints order

Make a selection from the **Predefined elements** list. The predefined elements vary depending on the application mode. You can also create a custom element by editing the shape functions and integration and constraint orders. The **Predefined elements** list then shows no selection.

For information about the finite element types see “Finite Elements” on page 482 of the *COMSOL Multiphysics Reference Guide*.

Specifying Weak Terms

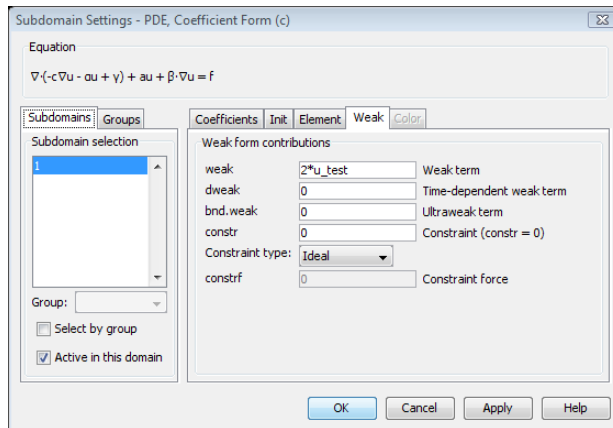


Figure 4-22: The weak term settings in the Subdomain Settings dialog box.

To specify additional weak terms on selected subdomains click the **Weak** tab. For more information on modeling using weak form contributions see the chapter “The Weak Form” in the *COMSOL Multiphysics Modeling Guide*. You can also specify additional pointwise constraints in the **constr** edit field (these are not weak constraints).

Ideal and Non-Ideal Constraints

The application modes support two types of constraints, ideal and non-ideal constraints. You can choose which constraint type to use in the **Application Mode Properties** dialog box on the **Physics** menu, except in the PDE modes, where you select the constraint type on the **Weak** page in the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, or **Point Settings** dialog box.

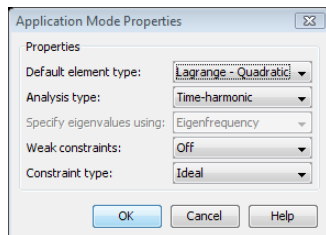


Figure 4-23: The Application Mode Properties dialog box with the constraint type setting.

The default constraint type is *ideal*, which means that the application modes derive the constraint force directly from the constraint at a boundary, edge, or point. You can also use the *non-ideal* constraints, which for constraints written as $0 = R$ on $\partial\Omega$, changes the constraint force according to

$$\left(\frac{\partial R}{\partial u}\right)^T \mu \rightarrow -I_{\partial\Omega}\mu$$

where $I_{\partial\Omega}$ is an identity operator for points on the boundary $\partial\Omega$. For example, using two variables u_1 and u_2 and the constraints

$$\begin{cases} 0 = R_1(u_1, u_2) & \text{on } \partial\Omega \\ 0 = R_2(u_1, u_2) & \text{on } \partial\Omega \end{cases},$$

then the non-ideal constraint force is $-\mu$ acting on both equations for the boundary $\partial\Omega$.

Viewing and Modifying the Full Equation System

In the equation system view you can view and modify the full system of equations for any model.

SPECIFYING PDE COEFFICIENTS FOR THE EQUATION SYSTEM

To open the **Subdomain Settings** dialog box for the full equation system:

- I** Go to the **Physics** menu, point to **Equation System**, and then click **Subdomain Settings**.

2 In the **Subdomain Settings** dialog box edit the PDE coefficients for the equation system:

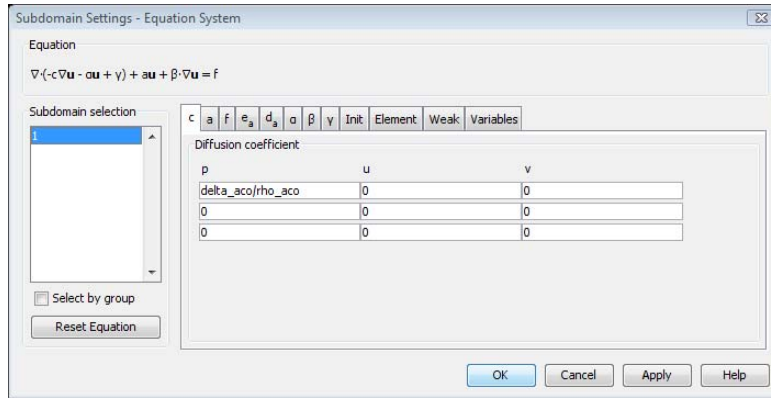


Figure 4-24: The Subdomain Settings dialog box for the full equation system.

The dialog box in Figure 4-24 shows the equation-system view for a model using the Acoustics and Plane Stress application modes. To keep track of the dependent variables and the application modes in the full equation system, the names of the dependent variables appear above the coefficient columns. This way you can relate the components of the system coefficients to the application modes and variables in a model. If any of the equations are deactivated in the selected subdomains, the corresponding rows and columns are disabled.

You can choose to display the equation system in three different forms: coefficient form, general form, or weak form. Select which form you want to display in the **Model Settings** dialog box; see page 186.

You can also view the equation system settings for boundaries, edges, and points by selecting the corresponding dialog box in **Physics>Equation System**.

The Differentiate Button

If the equation system form is the general form, the dialog box for the full equation system contains a **Differentiate** button in the lower-left corner that updates the coefficients c , α , β , and α for all subdomains to take into account any modifications you have made to the γ and f terms in this dialog box; as is indicated in the **Equation** area, COMSOL Multiphysics computes c , α , β , and α by differentiating γ and f . However, the software only uses the expressions in the edit fields for c , α , β , and α if, in addition, the solution form is the general form. By default, the solution form is the

weak form in most situations, but there are a few exceptions so it is good practice to click **Differentiate** if you make changes to the values in the edit fields for γ or f .

SPECIFYING INITIAL CONDITIONS FOR THE EQUATION SYSTEM

Click the **Init** tab to specify initial values for all dependent variables (and the first time derivatives of these variables) in the equation system on the selected subdomains.

SPECIFYING WEAK TERMS FOR THE EQUATION SYSTEM

Click the **Weak** tab to specify weak terms for all equations in the equation system on the selected subdomains. You can also specify additional pointwise constraints in the **constr** edit field (these are not weak constraints).

SPECIFYING ELEMENT TYPES FOR THE EQUATION SYSTEM

Click the **Element** tab to specify shape functions, integration order, and constraints order for the equation-system variables on the selected subdomains. For more information of the element types see “Finite Elements” on page 482 of the *COMSOL Multiphysics Reference Guide*.

VIEWING AND MODIFYING THE APPLICATION MODE VARIABLES

Click the **Variables** tab to see a list of the application mode variables in a model. They all appear with the following information:

- The variable name
- The expression that defines the variable
- A description of the variable

You can edit the expression to modify predefined application mode variables.

These application mode variables also appear in some of the postprocessing and visualization tools. See the documentation for the application modes in the *COMSOL Multiphysics Modeling Guide* for a complete list of variables.

Interior Mesh Boundary Variables

Some application modes define variables on interior mesh boundaries. If any of the application modes in the model define such variables, an additional tab, **Interior Variables**, appears in the dialog box, where you can view and edit these variables.

LOCKED SUBDOMAIN SETTINGS

If you change any PDE coefficients or other settings in the **Subdomain Settings** dialog box for the equation system, the original settings remain in the **Subdomain Settings**

dialog boxes for the application modes. However, COMSOL Multiphysics *locks* them for any subdomains where you have changed them at the system level. COMSOL Multiphysics then uses the settings for the equation system. The software indicates locked subdomains in the **Subdomain selection** lists with a padlock symbol. To reset the equation system to the values based on the settings for the application modes, click the **Reset Equation** button in the **Subdomain Settings** dialog box for the equation system.

Modeling Anisotropic Materials

The c coefficient in the coefficient form PDE corresponds to physical properties that can be anisotropic such as:

- Diffusion coefficient
- Permittivity
- Thermal conductivity
- Electric conductivity

For an anisotropic material, the c coefficient is a tensor with at most four components in 2D and at most nine components in 3D:

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

When you click in the edit field for the c coefficient— as well as for certain material properties in the physics modes—it expands to a matrix input for the tensor components with an associated drop-down list. In the list you can choose from the following material models:

- **Isotropic** (the default)—enter only one value.
- **Anisotropic - diagonal**—enter the diagonal components for an anisotropic material with the main axes aligned with the model’s coordinate system.
- **Anisotropic - symmetric**—enter a symmetric matrix using the diagonal components and the upper off-diagonal components.
- **Anisotropic - full**—enter the full 2-by-2 (2D) or 3-by-3 (3D) matrix for an anisotropic material.

To close the matrix-input pop-up window, press Esc.

You can also enter the components of an anisotropic coefficient as space-separated entries in the edit field in the dialog box. Notice the order of the components for a c coefficient in a 3D model:

- Diagonal anisotropic material: c_{11} c_{22} c_{33}
- Symmetric anisotropic material: c_{11} c_{12} c_{13} c_{22} c_{23} c_{33}
- Full anisotropic material: c_{11} c_{21} c_{31} c_{12} c_{22} c_{32} c_{13} c_{23} c_{33}

The ordering of components is similar in 2D.

Note: The Structural Mechanics Module and the MEMS Module support any anisotropic elasticity modulus defined in global or local coordinates.

Using the Model Tree

The Model Tree Views

The Model Tree, which appears to the left in the COMSOL Multiphysics user interface, provides three different views that provide access to the entire model. You select among these views using the toolbar buttons located at the very top of the Model Tree window.

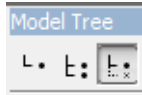


Figure 4-25: The buttons for the Model Tree views: Overview (left), Detail (center), and Inspect (right).

- The **Overview** button provides an overview of geometries and application modes in the model.
- The **Detail** button brings up a detailed view of the model including settings and groups at all levels as well as constants, expressions, coupling variables, and other settings.
- The **Inspect** button provides a detailed view that includes only those settings that the model uses. Constants, expression variables, coupling variables, function definitions, ODEs and space-independent equations, and changes at the equation-system level appear in this view only if you have used these features. This makes it easier to find the actual settings that the model uses.

In each of these three views, the Model Tree displays the corresponding data in two areas or subwindows:

- The top area contains the tree structure where you can navigate to select a part of the model and access dialog boxes and commands related to that part of the model.
- The bottom area displays information about the part of the model that you have selected in the top area.

It is possible to change the sizes of these two areas; with the mouse drag the border between the top area and the bottom area and the border to the right of the Model Tree.

The following sections provide details about the three views. The figures show the look of the Model Tree for the model “Thermal Effects in Electronic Conductors”

(documentation for that model starts on page 36 the *COMSOL Multiphysics Quick Start and Quick Reference*).

Using the Overview View

When you select the Overview, the Model Tree has the following look:

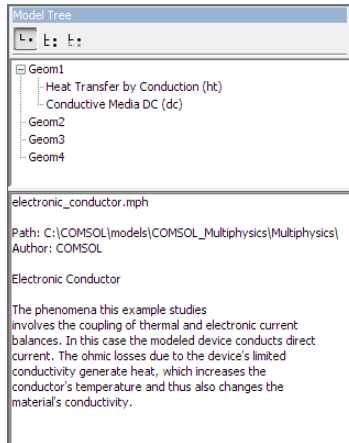


Figure 4-26: The Model Tree displaying a model in the Overview view.

Your first option is to click on a geometry, such as **Geom1** in Figure 4-26. (Note that any click on a geometry in the Model Tree also selects that geometry in the graphical user interface.) Selecting a geometry in the Model Tree displays its space dimension, the dependent variables for all application modes in that geometry, the independent variables (spatial coordinates), and the base unit system.

Next, by right-clicking on any geometry you can access associated dialog boxes for defining and editing expression variables in the domain (scalar, subdomain, boundary, edge, and point conditions). From this view you can also merge model with the current one. It is also possible to delete the geometry.

Finally, double-clicking a geometry opens the **Model Settings** dialog box where you can change the model settings in that particular geometry.

The next level down from a geometry shows the corresponding application modes. Clicking an application mode (**Heat Transfer by Conduction**, for example) displays its dependent variables and the application mode properties (for example, the analysis type). Right-clicking on an application modes provides access to all dialog boxes for

physics settings, application scalar variables (if applicable), and application mode properties. It is also possible to remove an application mode in this view.

Using the Detail View

When you select the Detail view, the Model Tree has the following look:

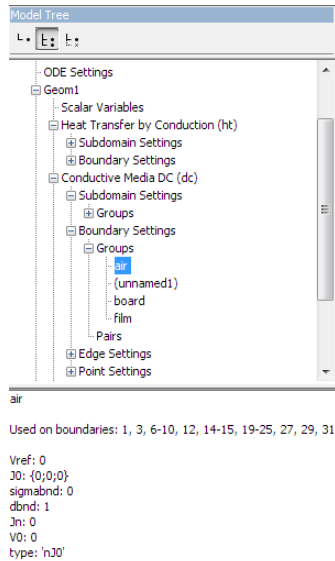


Figure 4-27: The Model Tree displaying a model in the Detail view.

Here you can access the model's settings at all levels:

- Double-click the top node to open the **Model Properties** dialog box. Right-click the same node to save the model, open the Model Navigator, add a model to the current model, or define constants, global expressions, functions, and ODEs.
- Double-click or right-click on **Constants**, **Global Expressions**, **Functions**, or **Global Equations** to open the corresponding dialog box.
- To see what is possible to do at a geometry folder's top level, see the previous section, "Using the Overview View" on page 224.
- In the subentries for a geometry, click **Scalar Variables** to display the application scalar variables (if applicable). Double-click or right-click to open the **Application Scalar Variables** dialog box.

- To see what is possible to do at an application mode folder's top level, see the previous section, "Using the Overview View" on page 224.
- At the next level of an application mode folder, the contents include the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings** folders (not all of these appear in 1D and 2D geometries). Click these folders to display the dependent variables and the number of domains of that type in the geometry. Double-click, right-click, or use the corresponding shortcut keys (F5–F8) to open the dialog boxes for these settings.

Open these folders to display groups of settings. Click a group to display its settings and the domains where it is used. Doing so also selects these domains in the graphical user interface. Right-click a group to copy the settings from that group, to paste a copied group setting to a group, and to open the corresponding settings dialog box. You can also delete unused groups.

For the boundary, edge, and point settings you can also access the settings for pairs that you have defined for an assembly. Click on a pair to display its settings, including the source domains and destination domains. Selecting any pairs also selects the source and destination domains. Right-click a group to copy and paste settings and to open the corresponding settings dialog box.

- The **Expressions** folder contains subentries for each type of expression variables in the domain (scalar, subdomain, boundary, edge, and point conditions). Right-click the **Expressions** folder to open any of the dialog boxes for these types of expressions. Click a subentry to display a list of expression variables of the corresponding type. Double-click or right-click a subentry to open the dialog box for that particular type of expressions. Selecting **Scalar Expressions** lists all scalar expression variables in the bottom area of the Model Tree. If a model contains expression variables in this geometry, you can open the corresponding folder, for example, **Subdomain Expressions**, and click on the individual groups to see which domains they include and the names and definitions for the expression variables in that group. Double-clicking or right-clicking provides access to the appropriate dialog box for the expression variables. By right-clicking on a group you can also copy and paste the expression variables from that group to another.
- The **Equation System** folder contains the settings for the underlying equation system. Open the folder to access the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings** folders (not all of these appear in 1D and 2D geometries), which refer to the same dialog boxes as the entries under **Physics>Equation System** on the Main menu. You open the dialog boxes by double-clicking or right-clicking these folders. Normally you do not need to make any changes at the equation-system

level, but these settings also include lists of variables, which you can see in the Model Tree by opening one of these folders, then opening the **Variables** folder, and finally selecting one of the groups. By default, there is only one group because the variable definitions are the same in all domains. For some models there are no variables defined on edges and points.

- In the application mode folder you can also find the **Coupling Variables** folder, which in turn contains three entries: **Integration Coupling Variables**, **Extrusion Coupling Variables**, and **Projection Coupling Variables**. Right-click on any of these to open the corresponding dialog box for defining coupling variables in subdomains, on boundaries, or on points. By right-clicking **Extrusion Coupling Variables** you can also open the dialog boxes for periodic boundary conditions and identity boundary conditions.

Using the Inspect View

The Inspect view shows the same level of detail as the Detail view but without any of the settings that are empty or unchanged from their default values. This makes it easier to find the expression variables, coupling variables, user-defined functions, and equation modifications that a model contains.

For the introductory model, the Inspect view has the following look:

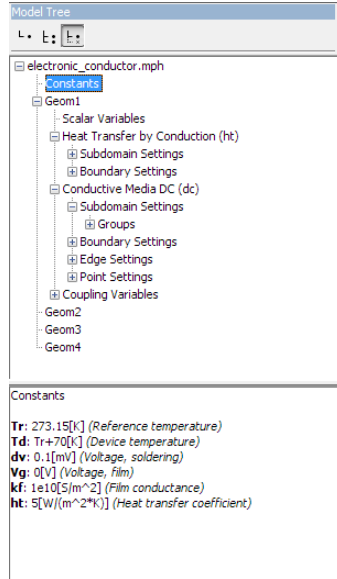


Figure 4-28: The Model Tree's Inspect view includes only those settings that the model uses.

This model uses constants, but there are no expression variables of any kind. There are also no ODEs or user-defined functions, so the Model Tree does not contain these parts. Moreover, this view does not include the **Equation System** folder because there have been no changes at that level.

Model Tree Settings

To remove the Model Tree, click the **Model Tree** button on the Main toolbar.

You can also change the preference settings for the Model Tree by choosing **Preferences** from the **Options** menu. In the **Model Tree** area select the **Show Model Tree** check box to include it in the user interface. Select the **Remember size** check box to use the specified size in subsequent COMSOL Multiphysics sessions. Select the **Show icons** check box to display the list with folders for list items that you can open to display additional details.

Using the Materials/Coefficients Library

The Material/Coefficients Library is a framework for a number of extensible databases with properties for a broad collection of physical, elastic, electromagnetic, fluid, piezoelectric, and thermal properties of materials as well as heat transfer coefficients. In the **Materials/Coefficients Library** dialog box you can add and edit library entries. Further, you can define multiple material libraries, each stored in a separate data file. The COMSOL Material Library is a separate add-on product that provides data, primarily as piecewise polynomial functions of temperature, for approximately 2500 materials. COMSOL Multiphysics and some of the modules include smaller libraries of material properties or coefficients. The number displayed next to a material library in the **Materials/Coefficients Library** dialog box indicates the number of materials in the library.

Using Material/Coefficient Data in Models

You have access to the COMSOL Multiphysics Materials/Coefficients Library from the **Subdomain Settings** dialog box in the physics modes and in the optional COMSOL modules. Selecting a material transfers the applicable properties to the current model. It is not necessary to define all the properties for an entry.

The available properties are grouped together in different groups according to their physical context. The available default groups are the Physics, Elastic, Electric, Fluid, Piezoelectric, and Thermal groups, each available on a separate tabbed page in the **Materials/Coefficients Library** dialog box. In addition, there is always a tabbed page named **All** that displays all properties. You can also add additional user defined groups with new or additional quantities together with a description of the quantity by editing the library data file. When loading materials from a domain settings dialog box, there is an additional tab with the same name as the tab in the domain settings dialog box, displaying only the properties that apply to the domain settings. The same property can appear in several groups with different descriptions in different groups. To view only

the defined material properties, select the **Hide undefined properties** check box in the **Materials/Coefficients Library** dialog box.

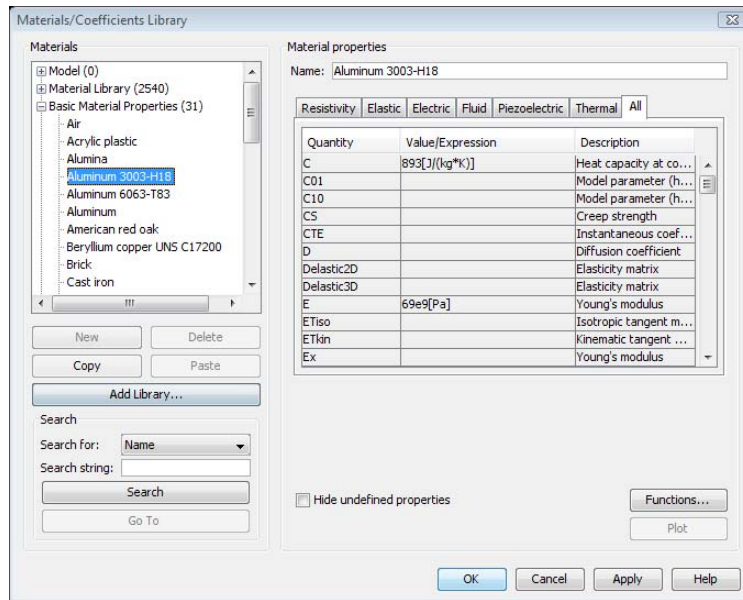


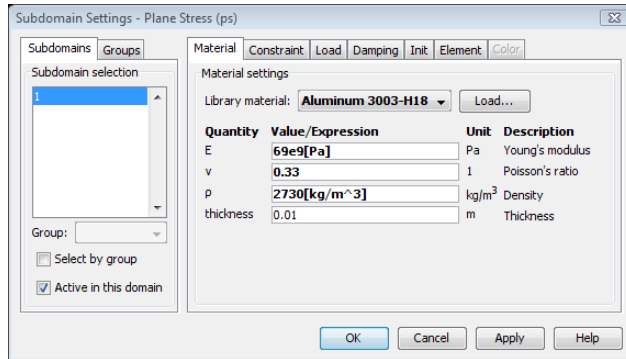
Figure 4-29: The Materials/Coefficients Library dialog box.

All Materials/Coefficients Libraries shipped with COMSOL Multiphysics and the optional modules are read-only. The **User Defined Materials** library is included as an example library to show how to add and edit your own materials. When you open the **Materials/Coefficients Library** dialog box from a physics-settings dialog box such as the **Subdomain Settings** dialog box, all materials including the ones in your model, the **User Defined Materials** library, and optional user-added libraries are read-only. To edit a material you must open the dialog box from the **Options** menu.

USING A LIBRARY MATERIAL OR COEFFICIENT

- 1 From the **Physics** menu open the **Subdomain Settings** dialog box.
- 2 Click the **Physics** or **Material** tab depending on the application mode.
- 3 Click the **Load** button to open the **Materials/Coefficients Library** dialog box.
- 4 In the **Materials** list, select the material you want to use in your model.
- 5 Click **OK**.

- 6 Go to the **Subdomain Settings** dialog box and select the desired item from the **Library material** list. COMSOL Multiphysics then uses the applicable material data in the selected subdomain. Data originating from the library appear in bold.



- 7 You can use different materials in different subdomains and edit the material data in the **Subdomain Settings** dialog box. Such edits affect only the current model.
- 8 When done, click **OK**.

If you change a value that originates from the Materials/Coefficients Library, COMSOL Multiphysics interprets this as meaning that you are no longer working with the material from the library. To indicate this, the selection of the material in the **Library material** list is canceled, and the values no longer appear in bold. If you change an argument to a function fetched from the Materials/Coefficients Library, this action is not considered as changing the material.

Instead of changing the subdomain settings after loading a material, you can edit the material directly in the Materials/Coefficients Library. COMSOL Multiphysics applies any changes you make in the material properties where the material is already used unless you add new coefficients or change the arguments of a function.

Note: Always review the material properties to make sure that they are appropriate for the model. For example, selecting **Air** provides temperature-dependent properties that are valid at pressures around 1 atm. Also pay attention to values for properties that are not included in the data for the selected material (data that do not appear in bold). You must specify the values for such properties manually.

Accessing Material Properties

The material properties that the application modes in the model use are typically available in the model. For example, the conductivity in a Conductive Media DC application mode with the name dc is always available as `sigma_dc`, regardless if the value comes from a material in the material libraries or not. You can also access any property value or property function from the materials that you have incorporated in a model by typing its name in the desired edit field. The names of the material properties use the local material names in the model as a prefix. For example, typing `mat1_rho` in any edit field provides the density of the material that you first incorporated in the model, while `mat2_rho` is the density for the second material. In the **Materials/Coefficients Library** dialog box, you find these prefixes in parentheses after the names of the materials in the **Model** folder (see Figure 4-30). In the example in that figure, `mat2_rho` is the density for air.

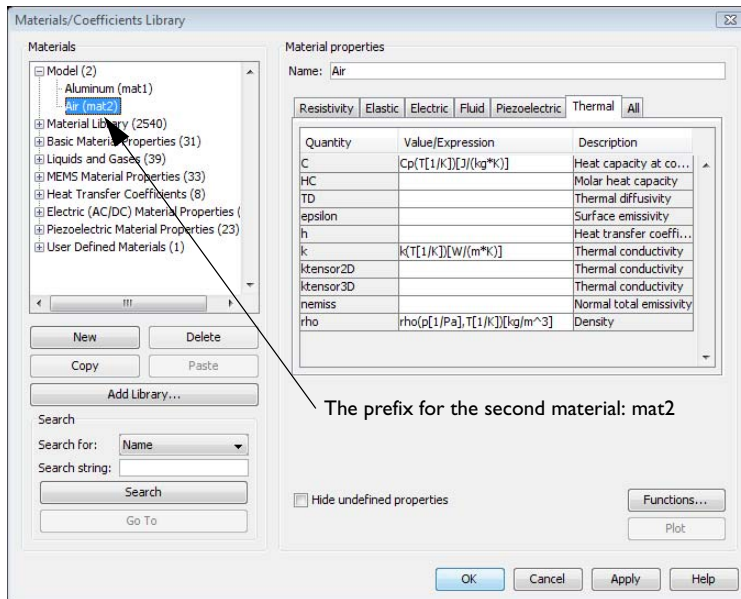


Figure 4-30: The prefixes for Aluminum (*mat1*) and air (*mat2*) appear next to the materials in the Model folder.

To extend the material data you can:

- Add new materials to the **User Defined Materials** database.
- Create your own material library from scratch.

ADDING AN ENTRY TO THE LIBRARY

- 1 Go to the **Options** menu and choose **Materials/Coefficients Library**.
- 2 In the **Materials/Coefficients Library** dialog box, select the **User Defined Materials** library in the **Materials**, or click the **Add Library** button to create a new library, or select an optional user defined library already available.
- 3 Click **New**.
- 4 Enter the name of the entry in the **Name** edit field under **Material properties**. By default, COMSOL Multiphysics names new entries **Material 1**, **Material 2**, and so on.
- 5 Enter values in the **Value/Expression** edit fields for the properties you want to define.
- 6 Click **Apply**.

You can also copy data from another library entry:

- 1 In the **Materials** list, select the library entry from which you want to copy data.
- 2 Click **Copy**.
- 3 In the **Materials** list, select the library entry to which you want to paste the data.
- 4 Click **Paste**.

USING FUNCTIONS IN A LIBRARY ENTRY

It is possible to define functions that are local to a material in the **Functions** dialog box, which you open by clicking the **Functions** button in the **Properties** frame of the desired material. This opens a dialog box that looks and acts the same as the **Functions** dialog box accessible from the **Options** menu except that it only displays functions local to the selected entry. See “Using Analytic Functions” on page 150 for more information on how to define functions.

For example, if $\text{rhofunction}(x)$ is a locally defined function for a material, you can set the quantity rho to $\text{rhofunction}(x)$. You can then specify the argument of the function when using the library entry in a domain settings dialog box such as the **Subdomain Settings** dialog box.

To access a local material function from somewhere else than the material, prefix the name with the material tag, such that a function `rhofunction(x)` in a material with the tag `mat1` can be referenced as `mat1_rhofunction(x)`.

CREATING A NEW LIBRARY

The best way to create a new library is to copy the existing library data file `data/mtrl_lib.txt` in the COMSOL installation directory and alter it according to your needs. You can also append library entries to an existing library data file. For information about the data structure for materials and coefficient libraries, see “Materials/Coefficients Libraries” on page 28 of the *COMSOL Multiphysics Scripting Guide*. The difference is that you in the library data files exclude the `fem` level and only specifies from the `lib` level and below. You can easily add new properties by just adding a new field. The example below shows how to add a new property `Eadd` to the existing Aluminum material in the Basic Material Properties library.

```
lib.mat{6}.name='Aluminum';
lib.mat{6}.variables.sigma='3.774e7[S/m]';
lib.mat{6}.variables.mur='1';
lib.mat{6}.variables.k='160[W/(m*K)]';
lib.mat{6}.variables.epsilonr='1';
lib.mat{6}.variables.C='900[J/(kg*K)]';
lib.mat{6}.variables.rho='2700[kg/m^3]';
lib.mat{6}.variables.nu='0.33';
lib.mat{6}.variables.alpha='23e-6[1/K]';
lib.mat{6}.variables.E='70e9[Pa]';
lib.mat{6}.variables.lambLame='5.1e10[Pa]';
lib.mat{6}.variables.muLame='2.6e10[Pa]';
lib.mat{6}.variables.lMurn='-2.5e11[Pa]';
lib.mat{6}.variables.mMurn='-3.3e11[Pa]';
lib.mat{6}.variables.nMurn='-3.5e11[Pa]';
lib.mat{6}.variables.Eadd='1.12e10[Pa]';
```

DEFINING GROUPS OF PROPERTIES IN A DATA FILE

You can define groups of parameters that appear on separate pages in the **Materials/Coefficients Library** dialog box. You define groups in the data files as, for example,

```
lib.groups{1}.name = 'Heat';
lib.groups{1}.variables = {'h'};
lib.groups{1}.descr = {'Heat Transfer Coefficient'};
```

The **Materials/Coefficients Library** dialog box then displays this group as a tab with the name **Heat**, and that group contains only the coefficient **h** with the description **Heat Transfer Coefficient**.

This is very handy to do when adding new properties who doesn't have a description to your data file.

ADDING A LIBRARY FILE TO THE DATABASE

- 1** Go to the **Options** menu and select **Materials/Coefficients Library**.
- 2** In the **Materials/Coefficients Library** dialog box click the **Add Library** button.
- 3** In the **Add Library** dialog box, browse to the library data file and select it.
- 4** Click **Open**.

A new library with the data in your file is now available in the **Materials** list.

ADDING A NEW EMPTY LIBRARY FILE TO THE DATABASE

- 1** Go to the **Options** menu and select **Materials/Coefficients Library**.
- 2** In the **Materials/Coefficients Library** dialog box click the **Add Library** button.
- 3** In the **Add Library** dialog box, enter the new name you want to give your library in the **File name** edit field and click **OK**.

A new empty library is now available in the **Materials** list.

DELETING A LIBRARY ENTRY OR LIBRARY

- 1** Select the library entry or library in the **Materials** list.
- 2** Click **Delete**.

Using MatWeb Material Property Data

MatWeb provides a service where you can export technical datasheets from MatWeb's collection in the format for a COMSOL material library. After downloading material property data from MatWeb, add this library file using the **Add Library** dialog box (see the previous section). For more information about this service, visit www.matweb.com.

Interpolation of External Data and Nonlinear Materials

Many material properties and other input data are not given by an analytical expression but can only be obtained by measurements. You can then use these measurements to approximate the material properties via interpolation in COMSOL Multiphysics. It is also possible to use other external data in your models.

The interpolated data can depend on solution variables (and their gradients) to model nonlinear materials. Example include approximating hysteresis curves for magnetic materials and temperature-dependent thermal properties such as thermal conductivity. Interpolation is usually performed in one dimension; that is, the material property is considered a function of one variable.

The easiest way to interpolate measured data is to create an interpolation function (lookup table). See “Using Functions Based on Interpolated Data” on page 154 for details. If you run COMSOL Script or MATLAB, you can also use interpolation functions in a command-line function. The example “A Rock Fracture Flow Model” on page 289 of the *COMSOL Multiphysics Model Library* illustrates both methods.

For an example of using external data, see “Example: Importing Unstructured 3D Data from an External Source” on page 240.

For nonlinear material models and nonlinear geometries in structural analysis, see the Structural Mechanics Module documentation.

Modeling Thermal Conductivity from Measurements

To illustrate the use of interpolation for nonlinear materials, create a model with a thermal conductivity that is based on measurements at different temperatures.

SETTING UP THE INTERPOLATION ROUTINE

The first step in setting up the interpolation routine is to define the values for the measured data. You can also load the data from a text file. The following table shows the measured data:

TABLE 4-27: THERMAL CONDUCTIVITY

TEMPERATURE	THERMAL CONDUCTIVITY
20	1.4
22	3.6
24	6.3
26	4.7
28	4.3
30	3.7
32	3.6
34	4.8
36	5.2
38	5.1
40	4.8

The following steps show how to enter this data set into the **Functions** dialog box:

MODELING USING THE GRAPHICAL USER INTERFACE—LOOKUP TABLE

- 1 Start COMSOL Multiphysics.
- 2 Select **2D** from the **Space dimension** list, and in the list of application modes select **COMSOL Multiphysics>Heat Transfer>Conduction>Steady-state analysis**. Click **OK**.
- 3 Draw a square with corners at (0, 0) and (1, 1).
- 4 Click the **Zoom Extents** button on the Main toolbar to adjust the coordinate system to the size of the rectangle.
- 5 Open the **Boundary Settings** dialog box. Select all boundaries, then in the **Boundary condition** list select **Temperature**. Type 20 in the **Temperature** edit field and click **OK**.
Define the interpolation function `therm_cond1(T)` by directly entering the conductivity data in Table 4-27.
- 6 From the **Options** menu choose **Functions**.
- 7 Click the **New** button.
- 8 In the **New Function** dialog box, go to the **Function name** edit field and enter `therm_cond1`, then click the **Interpolation** button. Click **OK**.

- 9 In the table, type the values for temperature and the thermal conductivity in the x and $f(x)$ columns, respectively.

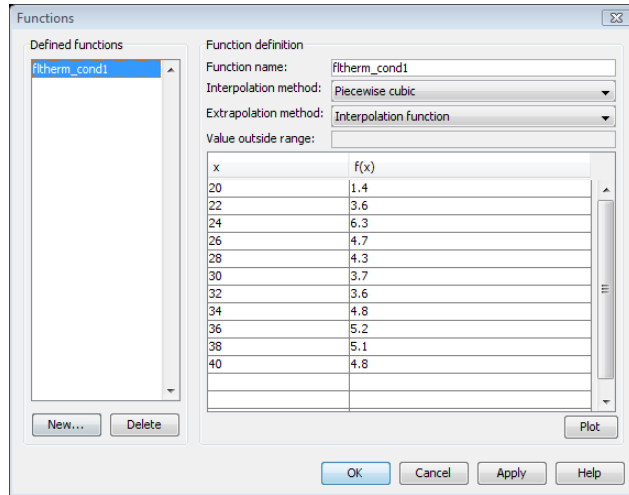


Figure 4-31: Interpolation function definition using a table in the Functions dialog box.

- 10 To plot the function, click the **Plot** button. The plot appears in a separate window.

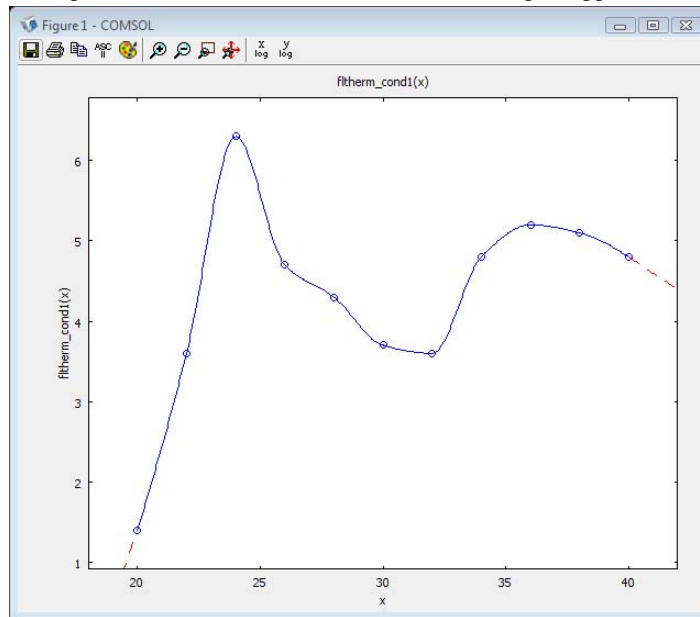


Figure 4-32: A plot of the interpolation function.

11 Click **OK**.

Next use this data in the specification of the thermal conductivity:

12 Open the **Subdomain Settings** dialog box and type `therm_cond1(T)` in the **Thermal conductivity** edit field. Type 750 in the **Heat source** edit field.

13 Click the **Init** tab and then type 20 in the **Temperature** edit field to set the initial value. Click **OK**.

14 Click the **Initialize Mesh** button on the Main toolbar to create a mesh.

15 Solve the problem by clicking the **Solve** button on the Main toolbar.

MODELING USING THE GRAPHICAL USER INTERFACE—DATA FROM FILE

In this case the same interpolation data is available as a text file, `thermcond.txt`, with the following format:

```
% Temperatures
20 22 24 26 28 30 32 34 36 38 40
% Thermal conductivity
1.4 3.6 6.3 4.7 4.3 3.7 3.6 4.8 5.2 5.1 4.8
```

Store a text file with this content as `thermcond.txt` somewhere on your file system. Then proceed as follows to retrieve the interpolation data from the file:

1 Follow Steps 1–7 from the previous section.

2 In the **New Function** dialog box, type `therm_cond1` in the **Function name** edit field and click the **Interpolation** button. Select **File** from the **Use data from** list.

3 Click the **Browse** button, browse to the directory where you saved `thermcond.txt`, and select this file.

4 Click **OK** to close the **New Function** dialog box.

5 In the **Functions** dialog box, select **Interpolation Function** in the **Extrapolation method** list to extrapolate the thermal conductivity outside of the specified temperature range. Click **OK**.

6 Open the **Subdomain Settings** dialog box and type `therm_cond1(T)` in the **Thermal conductivity** edit field. Type 750 in the **Heat source** edit field.

7 Click the **Init** tab and then type 20 in the **Temperature** edit field to set the initial value. Click **OK**.

8 Click the **Initialize Mesh** button to create a mesh.

9 Solve the problem by clicking the **Solve** button.

USING A POLYNOMIAL TO DESCRIBE THE DATA

If possible, a preferred method is often to fit a polynomial to your data. Once you have an analytic function for a nonlinear material (for example, $T = a + bT + cT^2$) you can enter that function directly using the edit field for the conductivity. Alternatively you can define the expression in the expression list. In that case, COMSOL Multiphysics takes care of the differentiation for the Jacobian contribution.

Example: Importing Unstructured 3D Data from an External Source

The procedure for importing external data on an arbitrary irregular computational mesh into a COMSOL model can be useful, for example, if you want to:

- Compare results from another program with solutions obtained in COMSOL Multiphysics. You can, for example, plot the imported data alongside the computation results from COMSOL Multiphysics.
- Use the imported external data as input data for further calculations in COMSOL Multiphysics.

This example demonstrates the second task by importing 3D velocity data stored in a text file, and then using it for further computations in a 3D convection and diffusion model.

The interpolation functionality in COMSOL Multiphysics supports different data formats. This example uses what is perhaps the simplest format: columns, where the leading columns contain coordinate data, and the following ones contain data points. The data is stored in a text file, whose structure is illustrated by the following excerpt:

```
% x          y          |... (u) [1/ (m/s)]          (v) [1/ (m/s)]          ...
0.0077083334 7.0E-4      ... -4.090323394685172E-18    2.408941441256009E-19    ...
0.007854166  7.0E-4      ... -1.6458371298236411E-18    3.7912376563092105E-20    ...
0.007854166  7.0E-4      ... -7.356721874724693E-18    9.05739114939885E-20     ...
0.0080       7.0E-4      ...  4.3662207463537125E-18    2.909759549458902E-18    ...
```

The % character indicates that the row contains headers.

This import example builds on the Model Library model in COMSOL_Multiphysics/Fluid_Dynamics/micromixer. You find the complete documentation for this model on page 246 of the *COMSOL Multiphysics Model Library*. Instead of computing the velocity field, you import velocity data from a file and use this data to compute the mass transfer problem.

MODEL NAVIGATOR

- I In the **Model Navigator**, go to the **Space dimension** list and select **3D**.

- 2 Select **COMSOL Multiphysics>Convection and Diffusion>Convection and Diffusion**.
- 3 Click **OK**.

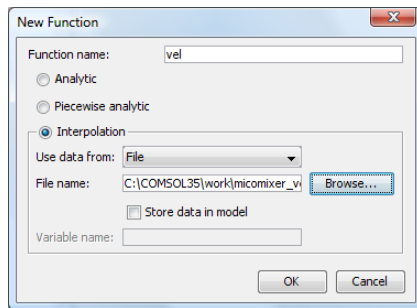
GEOMETRY IMPORT

To import the model geometry CAD file follow these steps:

- 1 From the **File** menu, select **Import>CAD Data From File**.
- 2 In the **Import CAD Data From File** dialog box, browse to the folder `models/COMSOL_Multiphysics/Fluid_Dynamics` in the COMSOL installation directory.
- 3 Select the file `micromixer.mphbin`, then click **Import**.

DATA IMPORT FROM FILE

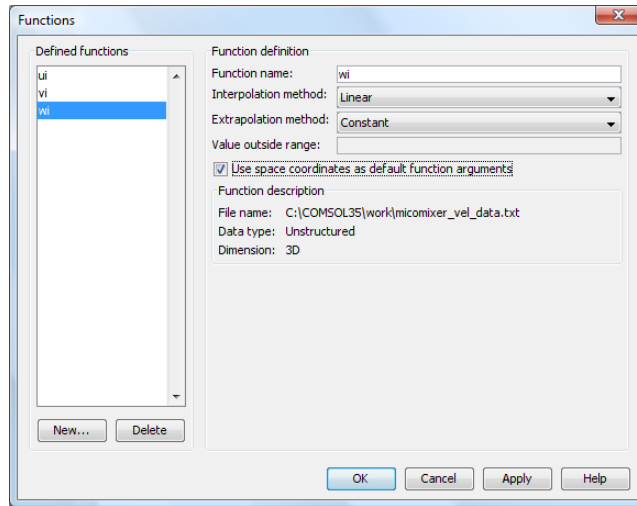
- 1 From the **Options** menu, select **Functions**.
- 2 Click **New**.
- 3 In the **New Function** dialog box, type `vel` in the **Function name** edit field.
- 4 Click the **Interpolation** button, then select **File** from the **Use data from** list.
- 5 Browse to the same Model Library folder as above and select the file `micromixer_vel_data.txt`. Click **Open** to load the file.



- 6 In the **New Function** dialog box, click **OK**.

Back in the **Functions** dialog box, you can inspect the interpolation function. COMSOL Multiphysics detects that the data file contains multiple columns with dependent variables and assigns each variable a new name by appending an index to your selected name (`vel1`, `vel2`, `vel3`). You can change a name by selecting the function in the list and typing the new name in the **Function name** edit field.

- 7 Rename the variables as u_i , v_i , and w_i ; for each variable, make sure to select the check box **Use space coordinates as default function arguments**. Click **OK**.



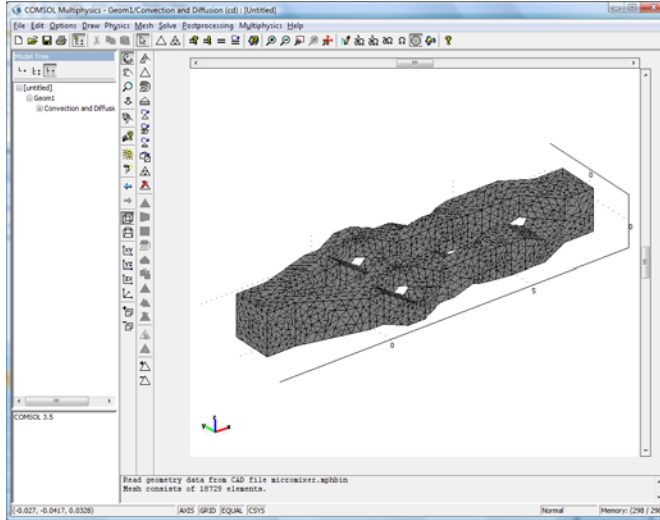
- 8 Click **OK** to close the **Functions** dialog box.

Now is a good time to visualize the imported data on the geometry. To do so, create a mesh and then overlay the functions on the mesh by updating the model from the **Solve** menu.

MESH GENERATION

- 1 From the **Mesh** menu, select **Free Mesh Parameters**.
- 2 On the **Global** page, click the **Custom mesh size** button.
- 3 Set the **Maximum element size scaling factor** to 0.25. This setting makes the element small enough to reasonably resolve the solution for this exercise.
- 4 Click **Remesh**.

5 When the mesher has finished, click **OK**.



6 From the **Solve** menu, select **Update Model**.

7 From the **Postprocessing** menu, select **Plot Parameters**.

8 On the **General** page, select the **Arrow** check box in the **Plot type** area (leaving the other settings at their defaults).

9 On the **Slice** page, type u_1 in the **Expression** edit field; this is the velocity component along the x -axis.

10 On the **Arrow** page, type u_1 , v_1 , and w_1 in the edit fields for **x-component**, **y-component**, and **z-component**, respectively. Change **x points** to 4, and select **3D arrow** from the **Arrow type** list.

11 Click **OK** to generate the plot and close the **Plot Parameters** dialog box.

The result on your screen should look like that in Figure 4-33 on page 244.

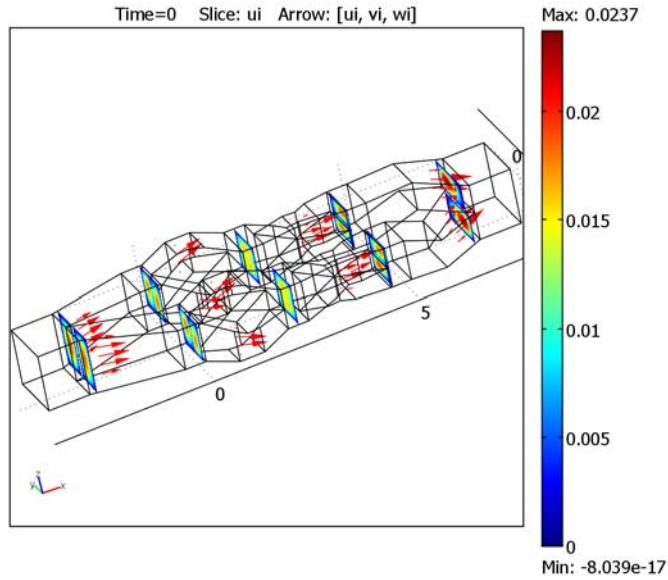


Figure 4-33: The velocity field imported from the data file.

OPTIONS AND SETTINGS

- 1 From the **Options** menu, select **Constants**.
- 2 Define the following names, expressions, and (optionally) descriptions:

NAME	EXPRESSION	DESCRIPTION
c0	27[mol/m ³]	Inlet concentration
D	1.49e-9[m ² /s]	Diffusion coefficient

- 3 Click **OK**.

PHYSICS SETTINGS—CONVECTION AND DIFFUSION

Boundary Conditions

- 1 Choose **Physics>Boundary Settings**.

2 Enter the following settings; when done, click **OK**.

SETTINGS	BOUNDARY I	BOUNDARY I36	ALL OTHERS
Boundary condition	Concentration	Convective flux	Insulation/Symmetry
c_0	$c_0^* (z < 0)$		

Subdomain Settings

- 1 From the **Physics** menu, open the **Subdomain Settings** dialog box.
- 2 Select all the subdomains, then enter coefficients from the following table, using the imported velocity data (which has an undefined unit):

NAME	EXPRESSION
D (isotropic)	D
R	0
u	u_i
v	v_i
w	w_i

- 3 Click the **Artificial Diffusion** button. In the dialog box that opens, select the **Streamline diffusion** check box. Click **OK**.
- 4 Click the **Init** tab. In the $c(t_0)$ edit field type c_0 .
- 5 Click **OK**.

COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar to compute the solution.

POSTPROCESSING AND VISUALIZATION

To visualize the concentration distribution, open the **Plot Parameters** dialog box, click the **Boundary** tab, activate **Boundary plot**, and select **Concentration, c** from the **Predefined quantities** list, before finally clicking **OK**.

Note: You can also use imported solutions as the dependent variables in an application mode. To do so, enter the variable names for the imported data in the corresponding edit field for the initial value on the **Init** page in the **Subdomain Settings** dialog box. Then choose **Get Initial Value** from the **Solve** menu.

Specifying Boundary Conditions

Boundary conditions define the interface between the model geometry and its surroundings. You can also set interface conditions on interior boundaries in a model geometry. For the physics modes and the optional COMSOL modules, select from predefined boundary conditions. You can set different boundary conditions for each boundary.

To specify general boundary coefficients for any COMSOL Multiphysics model use the **Boundary Settings** dialog box for the equation system. See “Viewing and Modifying the Full Equation System” on page 218 for more information about the equation system view.

Interior and Exterior Boundaries

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

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

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

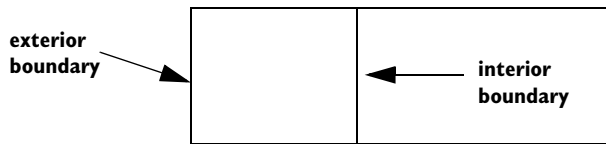


Figure 4-34: Examples of exterior and interior boundaries.

Setting Conditions on Boundaries

In the **Physics** menu, choose **Boundary Settings** to open the **Boundary Settings** dialog box. In this tabbed dialog box you select boundaries and specify boundary conditions.

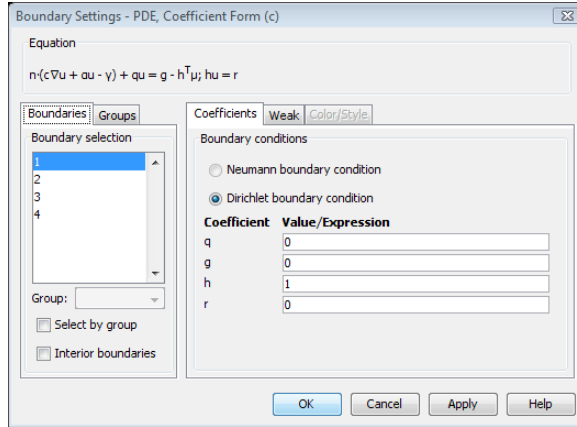


Figure 4-35: The Boundary Settings dialog box for a PDE, Coefficient Form application mode.

The equation at the top of the dialog box describes the currently selected boundary condition. A boldface **u** indicates a vector-valued equation with more than one dependent variable.

THE BOUNDARY CONDITIONS

Click the **Coefficients** tab to specify boundary conditions.

There are two basic types of boundary conditions:

- *Generalized Neumann condition (or Robin condition)*, where the boundary condition is determined by the coefficients q and g according to the equation

$$\mathbf{n} \cdot (c \nabla \mathbf{u} + \alpha \mathbf{u} - \gamma) + \mathbf{q} \mathbf{u} = \mathbf{g}.$$

The standard Neumann condition does not include the qu term.

In the system case with n dependent variables, q is an n -by- n matrix and g is an n -by-1 vector.

- *Dirichlet conditions*: u is specified on the boundary. The boundary condition equation is $h\mathbf{u} = \mathbf{r}$ where h is a weight that normally equals 1. In a model with n dependent variables, h is an n -by- n matrix and r is an n -by-1 vector. The Dirichlet condition also includes a Neumann condition:

$$hu = r, \quad \mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g - h^T \boldsymbol{\mu}.$$

The Neumann condition is active for the solution components where the h matrix does not impose a constraint (h is not full rank).

The physics modes contain predefined application-specific boundary conditions of both types.

For more information about boundary conditions for PDE modes and physics modes, see the *COMSOL Multiphysics Modeling Guide*.

You can define the coefficients in the boundary conditions with constants and expressions as well as mathematical and logical functions. For an overview of the available variables, see “Using Variables and Expressions” on page 138.

The default boundary condition for boundaries in the PDE modes is the Dirichlet condition $u = 0$. The default boundary condition for the physics modes varies.

The **Boundary Settings** dialog box for models with more than one dependent variable, built using the PDE modes, use a tabbed dialog box with separate pages for the q , g , h , and r coefficients and the weak terms. Figure 4-36 shows the **Boundary Settings** dialog box for a coefficient form PDE with two dependent variables:

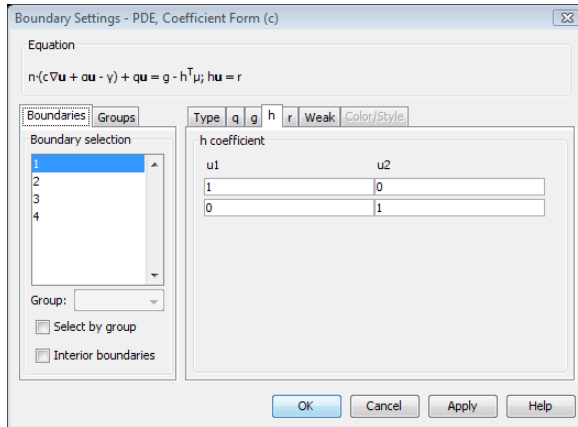


Figure 4-36: The *Boundary Settings* dialog box for a 2-variable PDE, Coefficient Form application mode.

THE WEAK TERMS

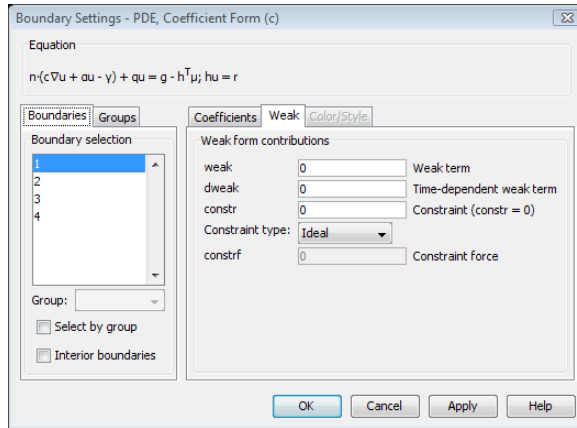


Figure 4-37: Weak term settings using the Boundary Settings dialog box.

Click the **Weak** tab to specify additional weak terms on selected boundaries. For more information on modeling using weak-form contributions, see the chapter “The Weak Form” in the *COMSOL Multiphysics Modeling Guide*. You can also specify additional pointwise constraints in the **constr** edit field (these are not weak constraints).

HANDLING CONFLICTING SELECTIONS

If you select multiple boundaries, the boundary settings apply to all the selected boundaries. If the boundary settings on the currently selected boundaries differ, the **Boundary conditions** button or list is *locked*, which COMSOL Multiphysics indicates with a yellow background color. To apply a new boundary setting for all selected boundaries, you must override the locked settings. To do so, enter the desired boundary settings, and the yellow background disappears.

Modifying Boundary Settings for the Equation System

To view and set the boundary conditions on the full equation system as standard boundary coefficients:

- 1 Go to the **Physics** menu, point to **Equation System**, and then click **Boundary Settings**.
- 2 In the **Boundary Settings** dialog box, edit the boundary coefficients for the equation system.

Here you can set boundary conditions for:

- Couplings between different application modes in a multiphysics model
- Boundary conditions not supported in the standard **Boundary Settings** dialog box for the application mode with which you are working

The application-specific boundary conditions are translated into standard boundary condition types and boundary coefficients for the full system. The names of the dependent variables appear above the coefficient rows so that you can relate the components of the system coefficients to the different application modes and variables in a model.

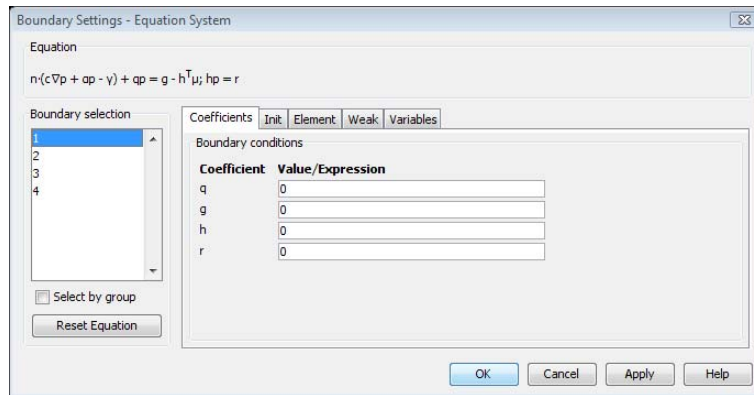


Figure 4-38: The *Boundary Settings* dialog box for the full equation system.

If the equation system form is the general form (see page 186), the dialog box contains a **Differentiate** button in the lower-left corner that updates the coefficients q and h for all boundaries to take into account any modifications you have made to the g and r terms in this dialog box; as is indicated in the **Equation** area, COMSOL Multiphysics computes q and h by differentiating g and r . However, the software only actually uses the q and h coefficients if, in addition, also the solution form is general form. By default, the solution form is weak form in most situations, but there are a few exceptions so it is good practice to click **Differentiate** if you make changes to the values in the edit fields for g or r .

SETTING INITIAL VALUES ON BOUNDARIES

Click the **Init** tab to specify initial values on the selected boundaries. It is normally not necessary to specify initial values on boundaries. However, it can sometimes be useful

if a variable is defined only on a boundary such as for Lagrange multipliers in weak constraints.

SPECIFYING WEAK TERMS FOR THE SYSTEM

Click the **Weak** tab to specify weak terms on boundaries for all variables in the equation system on the selected boundaries. You can also specify additional pointwise constraints in the **constr** edit field (these are not weak constraints).

In most circumstances the constraint force, which you find in the **constrf** edit field, should be equal to the constraint. Therefore if you modify the constraint you also need to modify the constraint force. This is true both if you enter the constraint using the h and r coefficients or the **constr** edit field.

SETTING ELEMENT TYPES ON BOUNDARIES

Click the **Element** tab to specify the shape function, integration order, and constraints order on the selected boundaries.

VIEWING AND MODIFYING APPLICATION MODE VARIABLES

Click the **Variables** tab to see a list of the application mode variables on a model's boundaries. This screen lists all such variables along with the following information:

- The variable name
- The expression that defines the variable
- A description of the variable

In some rare cases you might want to edit the expressions that define the application mode variables.

LOCKING BOUNDARY SETTINGS

If you change any boundary settings in the **Boundary Settings** dialog box for the equation system, the original boundary conditions are still present in the dialog boxes for the individual application modes. However, they are *locked* on the boundaries where they have been changed at the system level. COMSOL Multiphysics indicates locked subdomains with a lock symbol in the **Boundary selection** lists. To reset the boundary settings to the values based on those for the application modes, click the **Reset Equation** button in the **Boundary Settings** dialog box for the equation system.

Setting Conditions on Interior Boundaries

By default you cannot set interface conditions on interior boundaries. COMSOL Multiphysics always ensures continuity of flux across interior boundaries, so in most cases there is no need to think about these conditions.

To enable the specification of interface conditions on interior boundaries:

- 1 Go to the **Physics** menu and choose **Boundary Settings**.
- 2 Select the **Interior boundaries** check box.

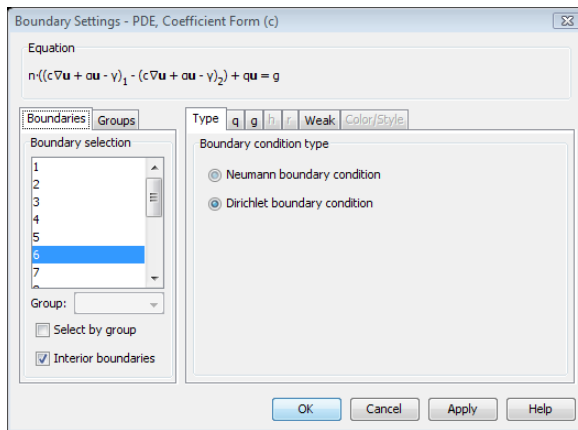


Figure 4-39: Activating interior boundaries in the Boundary Settings dialog box.

You can now select all boundaries from the **Boundary selection** list.

Applications where interface conditions on interior boundaries can be useful include:

- Internal sources and sinks
- Discontinuities in flux across an interior boundary

For each application mode, the **Boundary condition** list shows the applicable condition types for interior and exterior boundaries. If you select both types of boundaries, the list contains the conditions common to all boundaries.

The Neumann boundary condition for a coefficient form PDE on an interior boundary is

$$n \cdot ((c\nabla u + \alpha u - \gamma)_1 - (c\nabla u + \alpha u - \gamma)_2) + qu = g,$$

and with a nonzero value for g , the equation defines a jump in the flux across the boundary. The subscripts “1” and “2” represent the domains on either side of the boundary.

Note: When using normal and tangential vectors on interior boundaries, take care to distinguish between the upside and downside of the boundaries. See “Direction of the Normal Component on Interior Boundaries” on page 172 for more information.

Specifying Boundary Conditions for Identity Pairs

The identity pairs connect parts of an assembly. You can specify boundary conditions for these pairs by first clicking the **Pairs** tab in the **Boundary Settings** dialog box. The default boundary condition for the pair boundaries provides continuity for the field variable, but you can also select a number of other applicable boundary conditions from the **Boundary condition** list. In some applications modes, this list includes special “slit” boundary conditions such as contact resistance in the Conductive Media DC application mode. For more information about physics settings for identity pairs, see “Specifying Physics Settings on Pairs” on page 423 in the *COMSOL Multiphysics Modeling Guide*.

Specifying Point and Edge Settings

Use the settings on points and edges to describe, for example, point and line sources.

Using Point Settings

In the PDE modes, you can specify weak terms for the selected points using the **Point Settings** dialog box. See “Implementing a Point Source” on page 285 in the *COMSOL Multiphysics Modeling Guide* for an example.

In some physics modes, the **Point Settings** dialog box contains predefined settings where you can enter point sources directly.

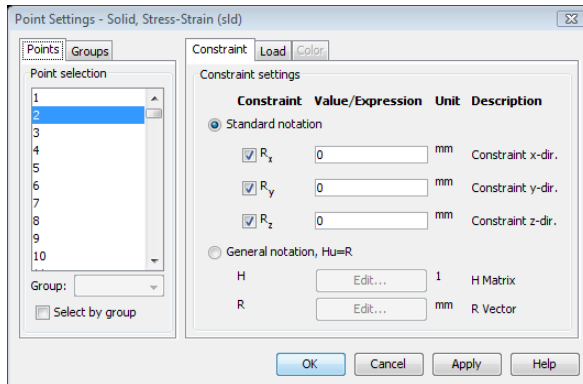


Figure 4-40: The Point Settings dialog box for a 3D Solid Mechanics application mode.

This dialog box is not available directly for all application modes.

POINT SETTINGS FOR THE EQUATION SYSTEM

To view and modify point settings for the full equation system:

- 1 Go to the **Physics** menu, point to **Equation System**, and then click **Point Settings**.

2 In the **Point Settings** dialog box enter weak contributions at points to create point sources and other pointwise contributions.

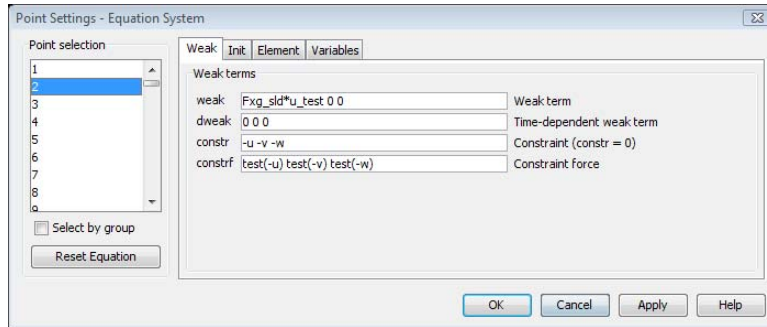


Figure 4-41: The Point Settings dialog box for the full equation system.

To specify initial values for points, click the **Init** tab.

To set element types and shape functions, click the **Elements** tab.

Using Edge Settings

Edge settings are available only for 3D models. They work in a similar way as the point settings. Specify edge settings, for example, to model line sources.

In the PDE modes you can specify weak terms for the selected edges using the **Edge Settings** dialog box. In some physics modes the **Edge Settings** dialog box contains predefined settings for line sources and other conditions on edges.

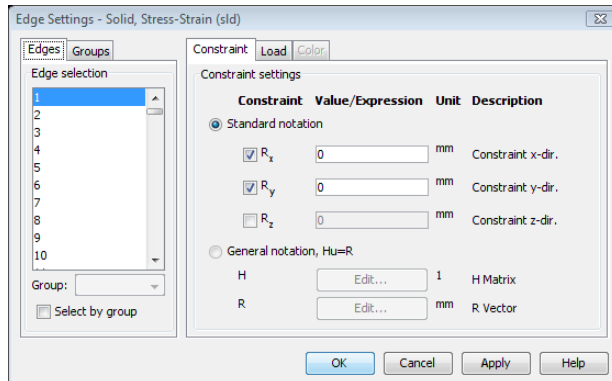


Figure 4-42: The Edge Settings dialog box for a Solid, Stress/Strain application mode.

This dialog box is not available directly for all application modes.

EDGE SETTINGS FOR THE EQUATION SYSTEM

To view and modify edge settings for the full equation system:

- 1 Go to the **Physics** menu, point to **Equation System**, and then click **Edge Settings**.

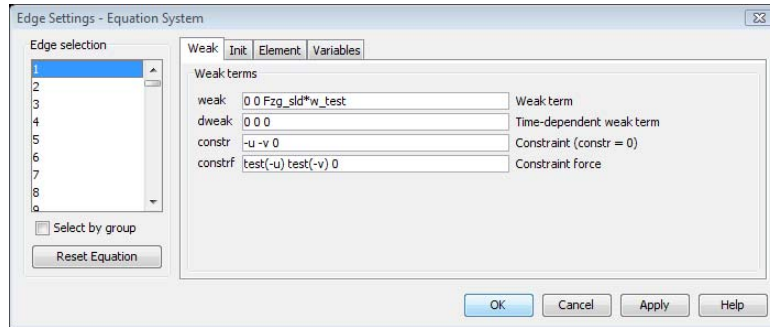


Figure 4-43: The Edge Settings dialog box for the full equation system.

- 2 In the **Edge Settings** dialog box, enter weak contributions on edges to create, for example, line sources.
- 3 To specify initial values on the edges click the **Init** tab.
- 4 To set element types and shape functions click the **Elements** tab.

Setting Point and Edge Constraints

You can define point and edge constraints using the **Point Settings** and **Edge Settings** dialog boxes for the equation system and directly using the PDE modes.

Use the **Constr** field to set a constraint. COMSOL Multiphysics constrains an expression written in the field to zero. For example, typing $0 \ v$ for a 2D structural mechanics model constrains the y -displacement v to zero.

Similarly, add a point load by typing $u_test * load$ in the **Weak** field, where u is the variable associated with the load. For an example of how to use a point load, see “Implementing a Point Source” on page 285 of the *COMSOL Multiphysics Modeling Guide*.

Using Periodic Boundary Conditions

Use *periodic boundary conditions*, *periodic edge conditions*, and *periodic point conditions* to define a constraint that makes two quantities equal on two different (but usually equally shaped) domains.

To implement a periodic boundary condition in COMSOL Multiphysics, go through the following steps:

- 1 Use a homogeneous Neumann boundary condition on all boundaries to which you want to apply the periodicity. This boundary condition normally corresponds to zero flux or insulation (see Table 4-28). When the boundaries are periodic, the inflow through one of the boundaries equals the outflow through the other.

TABLE 4-28: HOMOGENEOUS NEUMANN BOUNDARY CONDITION

PHYSICS MODE NAME	BOUNDARY CONDITION
Acoustics	Sound hard boundary (wall)
AC Power Electromagnetics	Electric insulation
Electrostatics	Zero charge/ symmetry
Magnetostatics	Electric insulation
Conductive Media DC	Electric insulation
Heat Transfer by Conduction	Thermal insulation
Convection and Conduction	Convective flux
Diffusion	Insulation/Symmetry
Convection and Diffusion	Convective flux
Structural mechanics application modes	No constraints, no loads (free)
Incompressible Navier-Stokes	Normal stress = 0

- 2 On the **Physics** menu, point to **Periodic Condition**, and then click the domain type: **Boundary**, **Edge**, or **Point**.
- 3 In the dialog box for the periodic condition, start by specifying the source geometry on the **Source** page. Type the expression that you wish to use in the periodic condition for the source. Refer to each constraint in the periodic boundary condition by a name. A default name appears automatically for the constraint that implements the periodic condition. To change the default name of the constraint

while working on the source page, type another name in the **Constraint name** column.

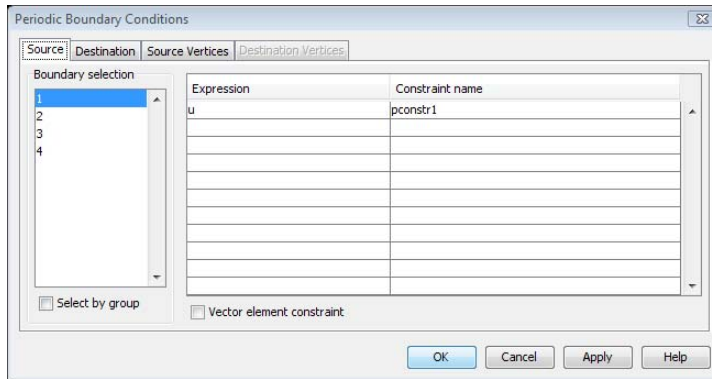


Figure 4-44: The Source page in the Periodic Boundary Conditions dialog box.

- 4 Click the **Destination** tab to define the destination domains. Select the geometry (if there is more than one geometry in the model) in the **Geometry** list and then click to select the boundaries that become the destinations. You are free to use several boundaries and more than one geometry. You can also select boundaries by clicking in the geometry and then selecting the **Use selected boundaries as destination** check box. Continue by selecting the constraint that you wish to specify in the **Constraint name** list. In the **Expression** edit field, type the expression that becomes equal to the expression on the source domain for the corresponding constraint; for a strictly periodic boundary condition, this is the same as the expression on the **Source** page.

Set the constraint order to the same order as the elements. This is the default setting, except when you are using application modes with different element orders.

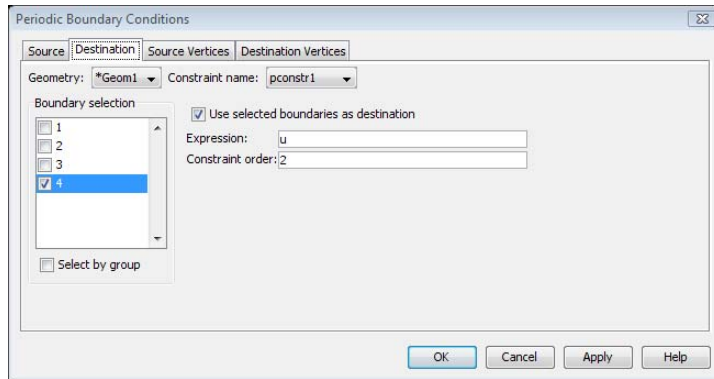


Figure 4-45: The destination page in the Periodic Boundary Conditions dialog box.

- 5 Click the **Source Vertices** tab to specify the transformation between the source and destination domains. Select 1, 2, 3, or 4 vertices that span the subspace constituted by the source domain. The number of vertices depends on the dimension of the source domain. For a periodic boundary, the number of vertices is the same as the dimension of the geometry. In a 2D simulation, the source vertices are typically the end points of the boundary.
- 6 Finally, click the **Destination Vertices** tab and select 1, 2, 3, or 4 vertices that span the subspace constituted by the destination domain; the number depends on the dimension of the destination domain, but you must provide the same number of points in both the source and destination domains. The source expression is projected onto this subspace. Its value at the first source vertex is projected to the first destination vertex, and so on. It is therefore important to assign the vertices in the correct order. Together the vertices on the source geometry and destination geometry define a linear transformation. See the following section for more information about the linear transformation from source geometry to destination geometry.

Note: When using periodic boundary conditions to connect boundaries or edges that do not have equivalent meshes, you might need to use the sparse null-space function, particularly in 3D. Normally the automatic selection of null-space function takes care of this. To specify the sparse null-space function manually, open the **Solver Parameters** dialog box from the **Solve** menu, click the **Advanced** tab, and select **Sparse** from the **Null-space function** list. Then click **OK**.

LINEAR TRANSFORMATION FROM SOURCE TO DESTINATION

You specify a linear transformation by giving a set of points in the source geometry and in the destination geometry. The order of the vertices is significant. COMSOL Multiphysics constructs a linear transformation from the source domain to the destination domain using the subspaces spanned by the vertices. Denote the transformation rank by n , denote the source vertices by x_0, x_1, \dots, x_n , and denote the destination vertices by x'_0, x'_1, \dots, x'_n . After padding the source and destination vertices' vectors with zeros as necessary, COMSOL Multiphysics solves the following matrix equation for a *transformation matrix* T and a *translation vector* V :

$$\begin{aligned}x'_0 &= Tx_0 + V \\x'_1 - x'_0 &= T(x_1 - x_0) \\&\dots \\x'_n - x'_0 &= T(x_n - x_0)\end{aligned}$$

Choose n such that the source domain can be embedded in a subspace of dimension $n - 1$. For example, when the source domain is a straight line it suffices to have two vertices; when it is a curve that can be embedded in a plane, you need three vertices. Four vertices are required for a source domain that is a curved surface or a curve that can only be embedded in 3D.

IMPLEMENTATION OF THE PERIODIC BOUNDARY CONDITIONS

COMSOL Multiphysics implements a periodic boundary condition as a special case of the extrusion coupling variable. In particular, it adds a variable name equal to the constraint name on the destination domains with a value equal to the constrained expression. The periodic boundary constraint works as an *ideal constraint*; that is, the derivatives of the constraint with respect to the solution help form the reaction forces. If you need more freedom in specifying how the reaction forces are applied, use the

extrusion coupling variable directly and specify the reaction force and constraint as weak contributions.

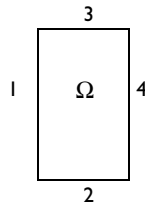
Periodic Boundary Condition Example

This example demonstrates how to implement a simple periodic boundary condition.

MODEL DEFINITION

Consider an eigenvalue equation on a rectangle of dimension $1 \times \sqrt{\pi}$.

$$\begin{aligned} -\Delta u &= \lambda u && \text{on } \Omega \\ u(x_1, y) &= u(x_4, y) && \text{on Boundaries 1 and 4} \\ u(x, y_2) &= u(x, y_3) && \text{on Boundaries 2 and 3} \end{aligned}$$



RESULTS AND DISCUSSION

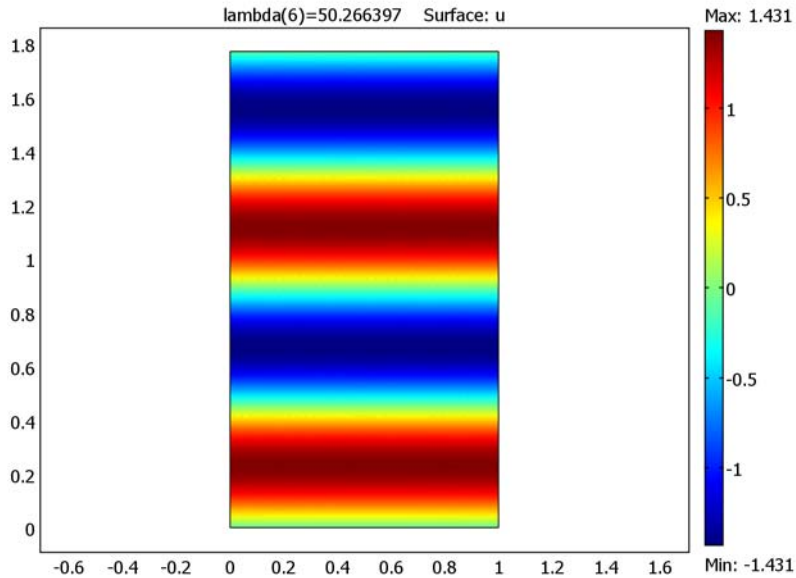


Figure 4-46: The periodic pattern in the solution (6th eigenmode).

The wavelength of the sinusoidal solutions is the rectangle side length divided by a series of integers, each of which corresponds to an eigensolution. The locations of the maxima are random because the model does not constrain the absolute value of the solution anywhere.

MODELING USING THE GRAPHICAL USER INTERFACE

Model Navigator

- 1 Start COMSOL Multiphysics.
- 2 In the **Model Navigator**, select **2D** from the **Space dimension** list.
- 3 In the **COMSOL Multiphysics>PDE Modes** folder select **PDE, Coefficient Form** and then **Eigenvalue analysis**. Make sure that **Lagrange - Quadratic** is selected in the **Element** list.
- 4 Click **OK**.

Geometry Modeling

- 1 Select the menu item **Draw>Specify Objects>Rectangle** to open the **Rectangle** dialog box. Type 1 in the **Width** edit field and $\sqrt{\pi}$ in the **Height** edit field.

- 2 Click **OK**.
- 3 Click the **Zoom Extents** button on the Main toolbar.

Physics Settings

Boundary Conditions

- 1 From the **Physics** menu open the **Boundary Settings** dialog box.
- 2 Select all boundaries and choose Neumann boundary conditions. Leave the default values.
- 3 Click **OK**.

Subdomain Settings

Use the default settings.

Periodic Boundary Conditions

- 1 On the **Physics** menu point to **Periodic Conditions**, and then click **Periodic Boundary Conditions**.
- 2 On the **Source** page select Boundary 1. Type u in the **Expression** edit field, then press **Enter**.
- 3 Click the **Destination** tab. Select the check box for Boundary 4. Type u in the **Expression** edit field.
- 4 Click the **Source Vertices** tab. Add Vertices 1 and 2 to the **Source vertices** list.
- 5 Click the **Destination Vertices** tab. Add Vertices 3 and 4 to the **Destination vertices** list.

This concludes the periodicity across the vertical boundaries. Proceed to the next steps to make the solution periodic across the horizontal boundaries.

- 6 Click the **Source** tab and select Boundary 2. On the second row, type u in the **Expression** edit field and press **Enter**.
- 7 Click the **Destination** tab and select the check box for Boundary 3. Type u in the **Expression** edit field.
- 8 Click the **Source Vertices** tab and add Vertices 1 and 3 to the **Source vertices** list.
- 9 Click the **Destination Vertices** tab and add Vertices 2 and 4 to the **Destination vertices** list.
- 10 Click **OK**.

Mesh Generation

- 1 Click the **Initialize Mesh** button on the Main toolbar.
- 2 Click the **Refine Mesh** button once.

Computing the Solution

The eigenvalue algorithm does not work reliably when the parameter that you specify in the **Search for eigenvalues about** edit field is equal to an eigenvalue. This particular problem has a zero eigenvalue, so you must change the value of this parameter.

- 1 Open the **Solver Parameters** dialog box.
- 2 Type 10 in the **Search for eigenvalues around** edit field.
- 3 Click the **Solve** button on the Main toolbar to compute the solution.

Postprocessing and Visualization

By default COMSOL Multiphysics plots the first eigenmode, which in this case is a flat solution associated with a zero eigenvalue.

- 1 From the **Postprocessing** menu open the **Plot Parameters** dialog box.
- 2 Look at the solution for different eigenvalues by making a corresponding selection in the **Eigenvalue** list. Start with the last eigenvalue.
- 3 Click **Apply** to plot each eigenvalue.
- 4 When finished click **OK**.

Specifying Application Scalar Variables

Some applications modes use scalar variables that take part in the mathematical model but are independent of the geometry and material. You can set the values of these variables in the **Application Scalar Variables** dialog box.

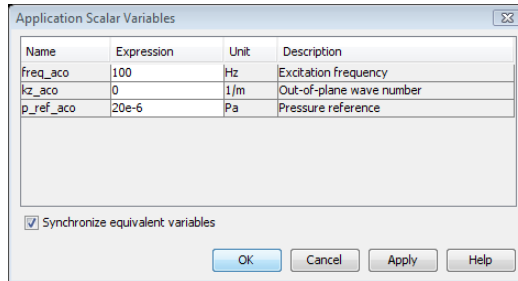


Figure 4-47: The Application Scalar Variables dialog box.

Go to the **Physics** menu and choose **Scalar Variables** to open the **Application Scalar Variables** dialog box.

You can change the value of each scalar variable by making an entry in the edit field in the **Expression** column.

The COMSOL Multiphysics application modes that include scalar variables are:

- *AC Power Electromagnetics*, which includes scalar variables for angular frequency, permittivity of vacuum, and permeability of vacuum
- *Electrostatics*, which includes a scalar variable for the permittivity of vacuum
- *Magnetostatics*, which includes scalar variables for the permittivity of vacuum and permeability of vacuum
- *Acoustics*, which includes scalar variables for the frequency and the reference pressure

Several other scalar variables appear in application modes defined in the add-on modules.

If your model does not contain any scalar variables, this menu item is not available.

Computing Accurate Fluxes

COMSOL Multiphysics provides two ways of computing accurate reaction forces and fluxes:

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

The model “Flow Past a Cylinder” on page 217 of the *COMSOL Multiphysics Model Library* shows how to use the reaction force operator to compute the lift and drag force.

Find further information on the use of weak constraints in the section “Using Weak Constraints” on page 358 of the *COMSOL Multiphysics Modeling Guide*.

When using weak constraints, the Lagrange multipliers are additional dependent variables in application modes using weak constraints (with variable names `lm1`, `lm2`, and so on). When using the reaction force operator, the reaction force operator of a certain dependent variable corresponds to the Lagrange multiplier of that dependent variable. The Lagrange multipliers correspond to the following postprocessing variables:

TABLE 4-29: PHYSICS MODES INTERPRETATION OF LAGRANGE MULTIPLIERS

PHYSICS MODE NAME	QUANTITY	CORRESPONDING VARIABLE
AC Power Electromagnetics	Surface current	$i \cdot \omega \cdot J_s$
Electrostatics	Surface charge density	nD
Magnetostatics	Surface current	$-J_s, -J_{sx}, -J_{sy}, -J_{sz}$
Conductive Media DC	Current flux	$-nJ$
Heat transfer application modes	Heat flux	$-nf_{lux}$
Diffusion application modes	Flux	$-nf_{lux}$

TABLE 4-29: PHYSICS MODES INTERPRETATION OF LAGRANGE MULTIPLIERS

PHYSICS MODE NAME	QUANTITY	CORRESPONDING VARIABLE
Structural mechanics application modes	Force	Tax, Tay, Taz
Incompressible Navier-Stokes	Total force per area	T_x, T_y, T_z

COMSOL Multiphysics computes only the part of the boundary flux captured by the Lagrange multiplier. You might have additional flux coming from boundary sources or nonidentity constraint matrices. This should not happen in the physics modes, though.

Flux Computation Example

Consider computing the flux over the constrained boundary of a rectangle. The PDE of the rectangle is Poisson's equation, $\Delta u = 1$, with $u = 0$ on the boundary. This example computes the flux using an expression, a reaction force operator, and a Lagrange multiplier.

MODEL NAVIGATOR

- 1 From the **Space dimension** list select **2D**.
- 2 In the **COMSOL Multiphysics** folder open **PDE Modes** and then **PDE, Coefficient Form**.
- 3 Click **OK**.

GEOMETRY MODELING

Draw a single rectangle of any size.

COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

POSTPROCESSING—FLUX EXPRESSION

- 1 On the **Postprocessing** menu choose **Subdomain Integration**.
- 2 In the **Subdomain Integration** dialog box type 1 in the **Expression** edit field.
- 3 Select Subdomain 1 and click **OK**. The rectangle's area appears in the message log.
- 4 Go to the **Postprocessing** menu and open the **Boundary Integration** dialog box.
- 5 Type $n_x*u_x+n_y*u_y$ in the **Expression** edit field.
- 6 Select all boundaries and click **OK**. COMSOL Multiphysics computes the flux over the boundaries as the normal component of the gradient of u .

The influx is approximately equal to the integral of the negative source term in Poisson's equation.

POSTPROCESSING—REACTION FORCE OPERATOR

- 1 Go to the **Postprocessing** menu and open the **Boundary Integration** dialog box.
- 2 Type $\text{react}(u)$ in the **Expression** edit field.
- 3 Select all boundaries and click **OK**. COMSOL Multiphysics computes the flux over the boundaries using the reaction force operator for the solution u .

PHYSICS SETTINGS—LAGRANGE MULTIPLIER

Application Mode Properties

- 1 Go to the **Physics** menu and choose **Properties**.
- 2 Select **On** from the **Weak constraints** list and then click **OK**.

COMPUTING THE SOLUTION—LAGRANGE MULTIPLIER

Click the **Solve** button on the Main toolbar.

POSTPROCESSING—LAGRANGE MULTIPLIER

- 1 Go to the **Postprocessing** menu and choose **Boundary Integration**.
- 2 Select **ImI** from the **Predefined quantities** list.
- 3 Select Boundary 1 and click **OK**. COMSOL Multiphysics computes the flux over the boundary by using the weak constraint variable (the Lagrange multiplier).

The integral of the flux exactly equals the integral of the finite-element projection of the source term in Poisson's equation.

Using Coupling Variables

Coupling variables are extremely powerful in their ability to make the value and the exact Jacobian of an expression available nonlocally. They are not only useful for modeling coupled problems—you can also use them solely for postprocessing and visualization purposes.

You define all coupling variables in two steps. First define the *source*, that is, which expression to evaluate and where to evaluate it; then define the *destination*, that is, the domains within which you want to use the resulting variable.

There are three major kinds of coupling variables available: integration, extrusion, and projection coupling variables. This section describes them all. For all types of coupling variables, always add each coupling variable on a separate row in the table on the **Source** page, even if they are defined on different domains.

Integration Coupling Variables

An *integration coupling variable* is the value of an integral of an expression over a set of domains. Thus, the integration coupling variable is always a scalar quantity. The source of the variable can be an expression at a vertex, or the integral of an expression over one or several subdomains, boundaries, or edges. The destination of the integration coupling variable can be the global destination, subdomains, boundaries, edges, or vertices.

Define an integration coupling variable by going to the **Options** menu and pointing to **Integration Coupling Variables**. Then select **Subdomain Variables**, **Boundary Variables**, **Edge Variables**, or **Point Variables**, depending on your intended source for the coupling variable.

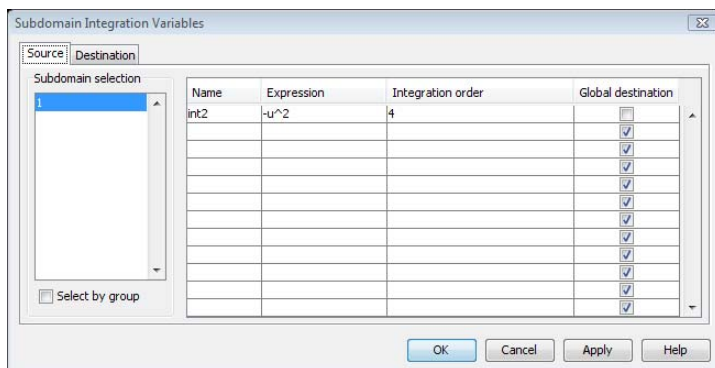


Figure 4-48: Specifying the sources for the subdomain integration coupling variables.

In the dialog box, select source domains for the integration coupling variables in the domain selection list—in the example above it is the **Subdomain selection** list—or by clicking on domains in the drawing area. Set the name of the coupling variable in the **Name** field, and then provide the expression to be integrated on these domains in the **Expression** field. The expression can contain any variables or functions defined on the domains. Also make an entry in the **Integration order** field, and optionally set the global destination (by default, the integration coupling variables are available globally, that is, on all domains).

You can tell on which domains a variable is defined by examining the **Expression** field. If the **Expression** field is empty, the variable is not defined on the selected domain(s). Select the **Select by group** check box to select all domains with the same **Expression** field entry. If you select multiple domains with different **Expression** field entries, the selection conflict is indicated by a yellow background color. If you enter an expression in the yellow **Expression** field, the new expression applies to all the selected domains.

Note: You can use the `dest` operator to force the expression to be evaluated on the destination side instead of the source side. See also “The `dest` Operator” on page 165.

To specify the destination of the coupling variable, clear the **Global destination** check box and then click the **Destination** tab. First select the destination geometry in the **Geometry** list. Then select the domain type in the **Level** list and select the check boxes for the domains in the list where you want the integration coupling variable to exist.

You can also select the domains directly in the drawing area and then select the **Use selected boundaries as destination** check box (or corresponding check box for other domain types). The expression on the source domains will be available on the selected destination domains. An asterisk in front of the name of a geometry or domain type (level) indicates that the integration coupling variable is available on that geometry and for that domain type.

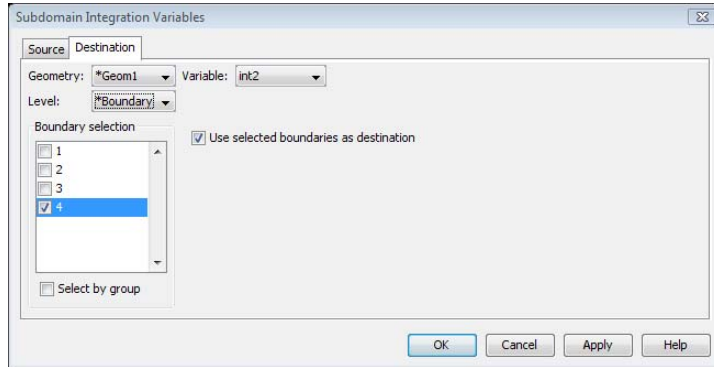


Figure 4-49: Specifying the destinations for the subdomain integration coupling variables.

If your variable name does not conflict with another variable at any domain, or if you would like to define the variables exactly where they are needed, use the global destination.

The following examples give some hints on how to use integration coupling variables.

INTEGRATING ON A DEFORMED MESH

When you are working with multiple frames (see “Frames for Deformed Meshes” on page 458 of the *COMSOL Multiphysics Modeling Guide*) the **Integration Coupling Variables** dialog box includes an extra **Frame** column where you can select the frame on which the expression should be integrated.

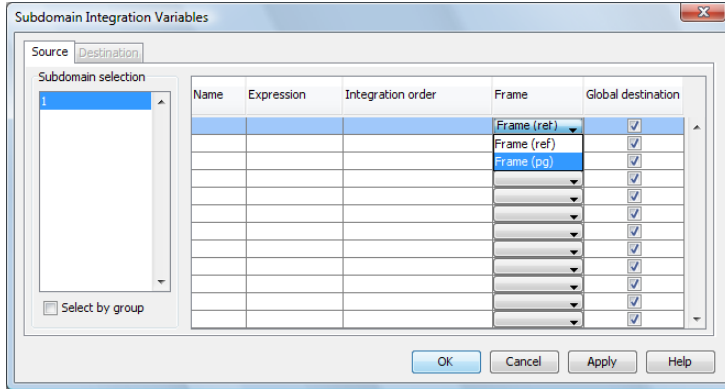
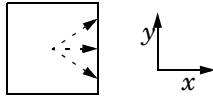


Figure 4-50: Specifying the frame for integration coupling variables.

Integration Coupling Variables—Examples



Consider Poisson's equation on a rectangular domain. The integral of the solution squared serves as the influx in a Neumann boundary condition on the right boundary. There is a Dirichlet boundary condition on the left boundary, and the top and bottom boundaries have zero influx.

$$\begin{aligned}
 -\Delta u &= 1 && \text{on } \Omega \\
 u &= x && \text{on } \partial\Omega_1 \\
 \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega_{2,3} \\
 \frac{\partial u}{\partial n} &= -\int_{\Omega} u^2 d\Omega && \text{on } \partial\Omega_4
 \end{aligned}$$

MODEL NAVIGATOR

- 1 In the **Model Navigator** and select **2D** from the **Space dimension** list.
- 2 In the **COMSOL Multiphysics>PDE Modes** folder, select **PDE, Coefficient Form**. Make sure **Lagrange - Quadratic** is selected in the **Element** list.
- 3 Click **OK**.

GEOMETRY MODELING

- 1 Draw a square of size 1 on a side by clicking on the **Rectangle/Square** button on the Draw toolbar, right-clicking on the origin, and then dragging to (1, 1).
- 2 Click the **Zoom Extents** button on the Main toolbar.

PHYSICS SETTINGS

Integration Coupling Variables

Set up an integration coupling variable that integrates the expression $-u^2$ over the rectangle and make it available on Boundary 4.

- 1 From the **Options** menu, choose **Integration Coupling Variables>Subdomain Variables** to open up the **Subdomain Integration Variables** dialog box.
- 2 Select Subdomain 1.
- 3 Enter the variable name `int2` and the expression `-u^2`. Use integration order 4 (the default). Clear the **Global destination** check box.
- 4 Click the **Destination** tab. Make sure Geometry 1 and the variable `int2` are selected. Select **Boundary** in the **Level** list, then select the check box for Boundary 4 in the **Boundary selection** list. This makes the variable `int2` available on Boundary 4.
- 5 Click **OK**.

A simple rule to keep in mind is that the integration order can be the maximum of the quadrature order for the integration of the shape functions of the variables in the expression.

Subdomain Settings

Use the default PDE coefficients (Poisson's equation). That is, do nothing.

Boundary Conditions

Change the following boundary coefficients; when done, click **OK**.

SETTINGS	BOUNDARY 1	BOUNDARIES 2, 3	BOUNDARY 4
Type	Dirichlet	Neumann	Neumann
r	0		
g		0	int2

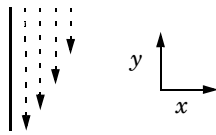
COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar to solve the problem.

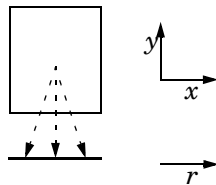
POSTPROCESSING AND VISUALIZATION

- 1 Go to the **Postprocessing** menu and select **Subdomain Integration**. Type $-u^2$ in the **Expression** edit field. Select Subdomain 1 and click **Apply**. The integral of the solution appears in the message log.
- 2 Open the **Plot Parameters** dialog box. On the **General** page, select the **Contour** check box and clear the **Surface** check box.
- 3 Click the **Contour** tab. On the **Contour Data** page, select **ux** from the **Predefined quantities** list. Click **OK**.
- 4 Click anywhere on the right boundary in the drawing area and verify that the influx is equal to the value of the integral.

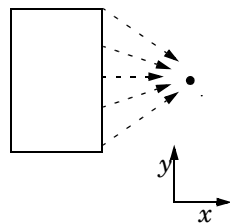
BRIEF EXAMPLES OF THE USE OF INTEGRATION COUPLING VARIABLES



One example of an integration coupling variable is when a scalar value from a vertex is used everywhere on a boundary to which the vertex belongs. In structural mechanics you can employ this type of coupling to formulate displacement constraints along a boundary in terms of the displacements of the end point. In electromagnetics the same technique can implement *floating contacts*.



Another example is to use the integral over a subdomain in a 2D geometry along a subdomain in another 1D geometry. This approach is helpful for process-industry models where two processes interact.



Finally, integration coupling variables can implement *integral constraints*. You first define a coupling variable at some vertex in such a way that it represents the value of the integral to be constrained. Then use a point constraint to set the coupling variable, and thereby the integral, to the desired value.

Extrusion Coupling Variables

An *extrusion coupling variable* maps values from the source domain to the destination domain. When the domains are of the same space dimension, you typically have a point-wise mapping. When the destination domain has higher dimension than the source domain, the mapping is done by extruding point-wise values to the higher dimensions. You can define the transformation between the source and destination in two ways: as a *linear transformation* or as a *general transformation*.

LINEAR TRANSFORMATION

The linear transformation maps between domains of the same dimension. The domains can exist in geometries of different space dimensions. For example, you can couple from edges in 2D to those in 3D; you can also couple 2D subdomains to 3D faces. In these cases you obviously need geometries of different space dimensions for the source and destination. You define the linear transformation by specifying points in both the source and destination.

To define an extrusion coupling variable with linear transformation, first go to the **Options** menu and point to **Extrusion Coupling Variables**. Then click **Subdomain Variables**, **Boundary Variables**, **Edge Variables**, or **Point Variables**, depending on the intended source for the coupling variable.

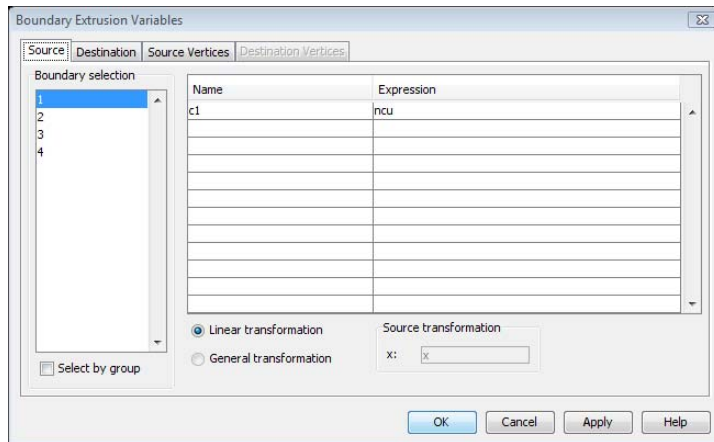


Figure 4-51: Specifying the sources for the boundary extrusion coupling variables (linear transformation).

In the dialog box, select source domains for the coupling variable in the domain selection list (in the previous example it is the **Boundary selection** list) or by clicking on

domains in the drawing area. Make an entry for the coupling variable in the **Name** edit field, and then provide an expression on these domains in the **Expression** edit field. Make sure that you have selected the **Linear transformation** button.

Now click the **Destination** tab. First select the geometry of the destination domain from the domains in the **Geometry** list. You can use the same geometry for both source and destination. The dimension of the destination domain is the same as the dimension of the source domain, regardless of the geometry dimensions. COMSOL Multiphysics transforms the expression on the source domains to the selected destination domains.

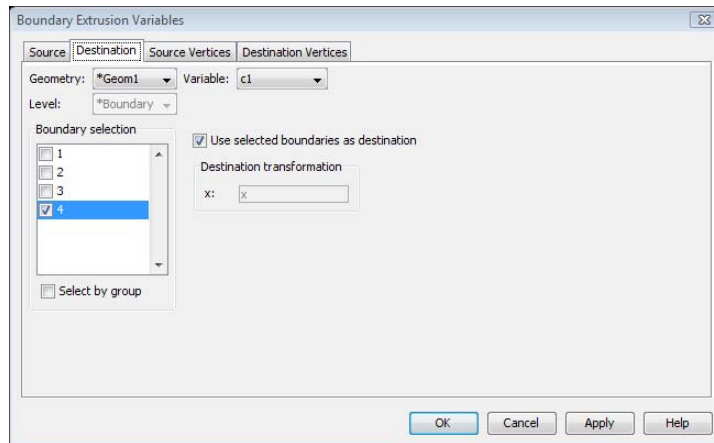


Figure 4-52: Specifying the destinations for the boundary extrusion coupling variables (linear transformation).

You can change the variable when working with the destination page. The default variable is the one selected on the source page.

Continue by specifying the linear transformation by giving a set of points in the source geometry and in the destination geometry. This works in exactly the same way as for periodic boundary conditions. See “Linear Transformation from Source to Destination” on page 260 for more information.

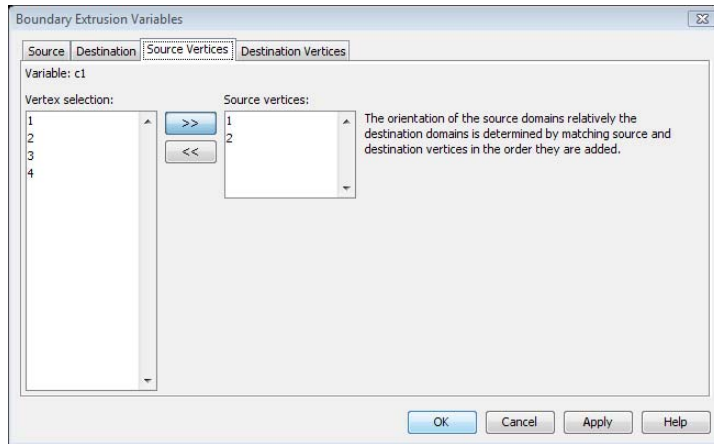


Figure 4-53: Specifying the source vertices for the boundary extrusion coupling variables.

Select vertices in the **Vertex selection** list and move them by clicking the right arrow button to the **Source vertices** or **Destination vertices** list, respectively.

Together, the vertices on the source geometry and destination geometry define a linear transformation.

GENERAL TRANSFORMATION

The *general transformation* of the extrusion coupling variable defines a more general transformation between source and destination than the linear transformation.

Specifically, when the destination domain has more space dimensions than the source domain, the variable performs extrusion of values.

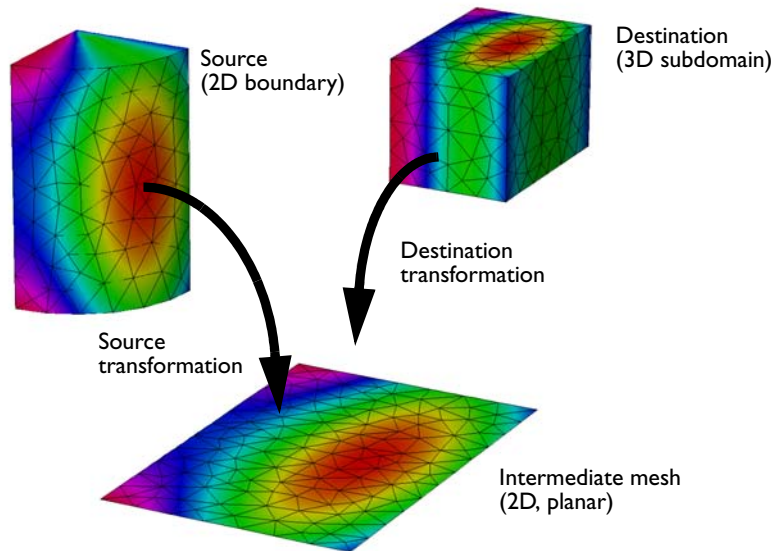


Figure 4-54: General transformation mappings.

The definition of any extrusion coupling variable involves two mesh transformations, which are important to understand. The *source transformation* is a one-to-one mapping that maps the mesh of the physical source domain to an *intermediate mesh* embedded in a space of the same dimension as the source. The *destination transformation* is a mapping from the destination domain, where the value of the variable is defined, to the same space that contains the intermediate mesh.

When you request the value of the coupling variable somewhere in the destination domain, the software transforms the destination points using the destination transformation. It compares the resulting coordinates to the elements in the intermediate mesh to find corresponding locations in the physical source domain. This means that the source transformation must be inverted but not the destination transformation. The latter can in fact be noninvertible, which is, for example, the case for a linear extrusion.

To avoid the need to solve a nonlinear system of equations for every destination point, the software assumes that the source transformation is linear on each element of the intermediate mesh. In practice, the transformation is often trivial and leaves the coordinates unchanged, but it can also rescale, stretch, bend, or reflect the mesh.

Choose the geometry of the source domain by opening the dialog box from that particular geometry. To define an extrusion coupling variable with general transformation, go to the **Options** menu and select **Extrusion Coupling Variable**. Then select **Subdomain**, **Boundary**, **Edge**, or **Point**, depending on the intended source for the coupling variable.

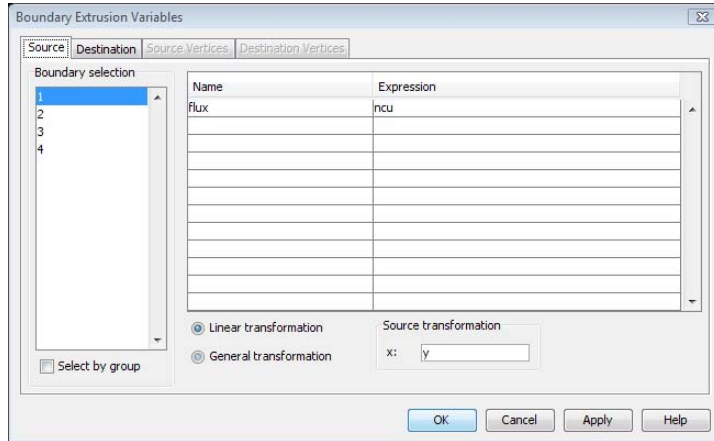


Figure 4-55: Specifying the sources for the boundary extrusion coupling variables (general transformation).

In the dialog box, select source domains for the coupling variable in the domain selection list; in the example above it is the **Boundary selection** list. Make an entry for the coupling variable in the **Name** edit field, and then provide an expression for the selected source domains in the **Expression** edit field. Make sure that you have selected the **General transformation** button.

Now define the source transformation that maps the source domain to the intermediate domain of the same dimension. The source transformation has the same number of fields as the dimension of the source domain. You can use expressions containing space coordinates in the source geometry when defining the transformation. Remember that the transformation must be approximately linear within each mesh element.

Now click the **Destination** tab. First select the **Geometry** and then select the domain type. You can change the variable when working with the destination page. The default variable is the one selected on the source page.

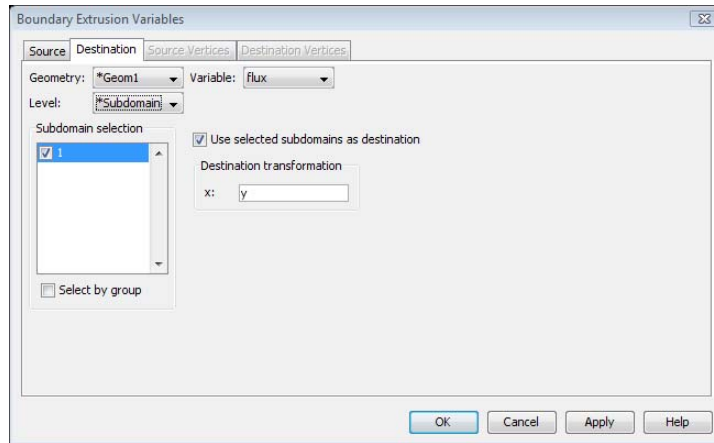


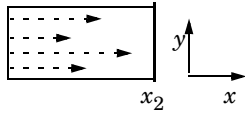
Figure 4-56: Specifying the destinations for the boundary extrusion coupling variables (general transformation).

Specify the destination transformation. The destination transformation maps coordinates in the destination domain to the intermediate mesh. The destination transformation has the same number of fields as the dimension of the source domain. When defining the transformation it is permissible to use expressions containing space coordinates in the destination geometry. You can also specify an arbitrary transformation, which can be highly nonlinear or noninvertible.

Extrusion Coupling Variables—Example

Consider the case of a single rectangular domain where the source term in Poisson’s equation comes from the inward flux over the right boundary for the corresponding y coordinate.

$$\begin{aligned} -\Delta u &= \frac{\partial}{\partial n} u(x_2, y) && \text{on } \Omega \\ u &= xy && \text{on } \partial\Omega \end{aligned}$$



The figure to the left illustrates the extrusion process. The values of the influx on the boundary become available throughout the domain by extrusion along the y -axis. The source transformation is y , and the destination transformation is y .

The problem is linear, so the linear solver is the one to work with. By using the weak term to specify the source term in Poisson's equation you can use the coefficient form as the solution form.

MODEL NAVIGATOR

- 1 In the **Model Navigator** select **2D** from the **Space dimension** list.
- 2 In the **PDE** folder, select **PDE, Coefficient Form**. Make sure **Lagrange - Quadratic** is selected in the **Element** list.
- 3 Click **OK**.

GEOMETRY MODELING

- 1 Draw a square with sides of length 1 by clicking on the **Rectangle/Square** toolbar button, right-clicking on the origin, and then dragging to (1, 1).
- 2 Press the **Zoom Extents** button on the Main toolbar.

PHYSICS SETTINGS

Boundary Conditions

- 1 Go to the Boundary Selection mode and select all boundaries.
- 2 From the **Physics** menu, choose **Boundary Settings** and type $x*y$ in the **r** edit field to specify the boundary condition $u = xy$. Click **OK**.

Subdomain Settings

- 1 In the **Subdomain Settings** dialog box, click the **Coefficients** tab and type 0 in the **f** edit field to set the source term to zero.
- 2 Click the **Weak** tab and type $u_test*flux$ in the **weak** edit field to specify a weak term in Subdomain 1.
- 3 Click **OK**.

You have entered Poisson's equation with the source $flux$. By using a weak term to specify the source term, the Jacobian matrix is accurate also for a nonlocal source term. The entire problem, including the coupling, is linear, so it suffices to use a linear solver to solve it.

Coupling Variables

- 1 Go to the **Options** menu, point to **Extrusion Coupling Variables**, and then click **Boundary Variables** to open up the like-named dialog box.
- 2 Select Boundary 1 and enter the variable name `f1ux` and the expression `ncu`.
- 3 Click the **General transformation** button. Type `y` as the source transformation in the `x` edit field.
- 4 Click the **Destination** tab. Select **Subdomain** in the **Level** list and then select Subdomain 1.
- 5 Type `y` as the destination transformation in the `x` edit field.
- 6 Click **OK**.

COMPUTING THE SOLUTION

To access the `ncu` variable, use the coefficient solution form:

- 1 From the **Solve** menu, choose **Solver Parameters**.
- 2 Click the **Advanced** tab.
- 3 Select **Coefficient** in the **Solution form** list.
- 4 Click **OK**.
- 5 Click the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

- 1 In the **Plot Parameters** dialog box, click the **Surface** tab.
- 2 On the **Surface Data** page, type `f1ux` in the **Expression** edit field.
- 3 Click the **Height Data** tab, select the **Height data** check box, and type `f1ux` in the **Expression** edit field.
- 4 Click the **Boundary** tab and then select the **Boundary plot** check box.
- 5 Type `ncu` in the **Expression** edit field on the **Boundary Data** page.

- Click the **Height Data** tab, select the **Height data** check box, and type `ncu` in the **Expression** edit field in the **Height data** area. Click **OK**.

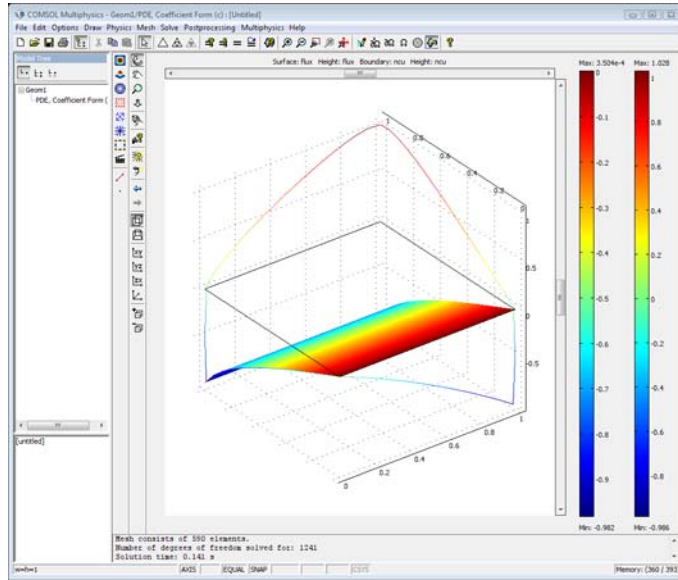
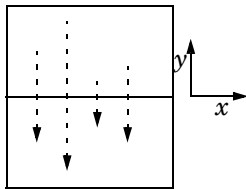


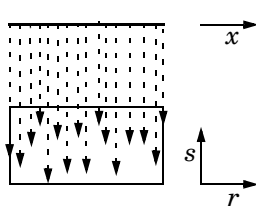
Figure 4-57: The flux (surface plot) and the `ncu` variable (boundary plot).

The plot in Figure 4-57 shows both the value of the `f1ux` variable on the destination domain and `ncu`, which is a variable that acts as the source on the boundary.

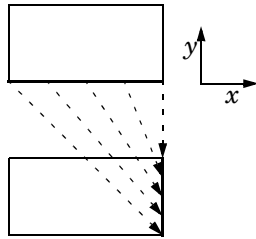
BRIEF EXAMPLES OF EXTRUSION COUPLING VARIABLES



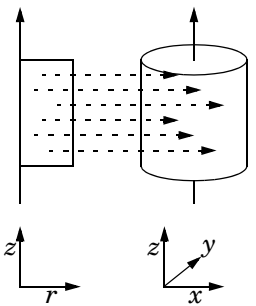
One application of extrusion couplings is to mirror the solution on the x -axis. This can be useful for postprocessing. The source transformation to enter is x , y , and the destination transformation is x , $-y$.



Another example is to extrude the solution in the 1D geometry to a 2D domain along the s -axis. The source transformation is x , and the destination transformation is r .



Yet another example maps values on the lower boundary of a rectangle that extends from $x = -1$ to $x = 1$ and from $y = 0$ to $y = 1$, to the right boundary on the same rectangle. The source transformation is $(x+1) / 2$ and the destination transformation is y .



Finally, to map the solution from a 2D axially symmetric model onto the original 3D domain, use the source transformation r , z and destination transformation $\sqrt{x^2+y^2}$, z . This mapping is particularly useful for postprocessing purposes. Note that if you revolve a 2D axisymmetric geometry in COMSOL Multiphysics, the axis of revolution in 3D is the y -axis by default. Then the destination transformation changes to $\sqrt{x^2+z^2}$, y (see “2D Axisymmetric Transient Heat Transfer” on page 196 in the *COMSOL Multiphysics Modeling Guide* for an example).

Projection Coupling Variables

A *projection coupling variable* evaluates a series of line integrals on the source domain, where the line positions depend on the positions of the evaluation points in the destination domain. In this way you can calculate the integral of an expression over one space variable for a range of different points along the other space axis, giving a result that varies over the latter space variable. For example, you can obtain the average along the y direction of a variable u defined on some 2-dimensional domain in the xy -plane by calculating the integral

$$\bar{u}(x) = \int u(x, y) dy$$

COMSOL Multiphysics uses a method whereby it first applies a one-to-one transformation to the mesh of the source domain. The last space dimension in the transformed mesh is the one integrated over, so the lines used to integrate are vertical in the transformed source mesh. The software takes the placement of the vertical lines in the transformed source mesh from the positions of the transformed destination evaluation points. It then carries out the integrals in the source domain over curves that correspond to the vertical lines in the transformed source mesh.

Then it applies a second transformation to the evaluation points in the destination domain, and it uses the resulting points or the interpolation of an expression at points in the transformed source mesh.

You can define the transformation between source and destination in two ways: as a *linear transformation* or as a *general transformation*.

LINEAR TRANSFORMATION

The linear transformation maps between source domains and destination domains of the nearest lower dimension. The domains can exist in geometries of different space dimensions. For example, you can couple from subdomains in 2D to edges in 3D; you can also couple 3D subdomains to 2D subdomains. Define the linear transformation by specifying points in both the source and destination domains.

To define a projection coupling variable with linear transformation, first go to the **Options** menu and point to **Projection Coupling Variables**. Then click **Subdomain Variables**, **Boundary Variables**, or **Edge Variables**, depending on the intended source for the coupling variable.

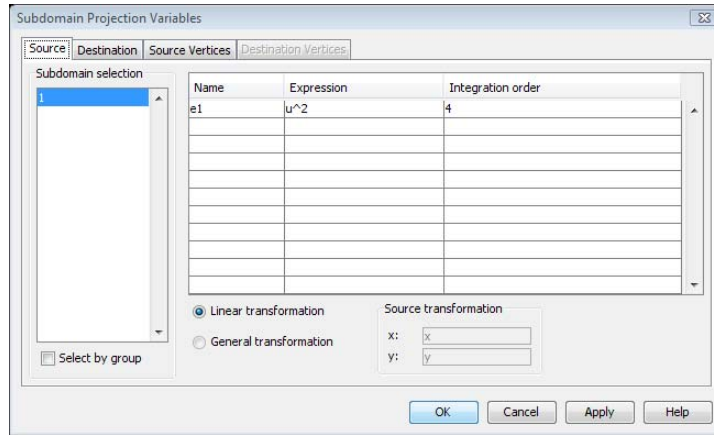


Figure 4-58: Specifying the sources for the subdomain projection coupling variables (linear transformation).

In the dialog box select the source domains for the coupling variable in the domain selection list (in Figure 4-58 it is the **Subdomain selection** list) or by clicking on domains in the drawing area. Make an entry for the coupling variable in the **Name** column, and then provide an expression on these domains in the **Expression** column. Also make an entry in the **Integration order** column. Make sure that you click the **Linear transformation** button.

Now click the **Destination** tab. First select the geometry of the destination domains in the **Geometry** list. The dimension of the destination domains is one level lower than the dimension of the source domains, regardless of the geometry dimensions. COMSOL Multiphysics transforms the expression on the source domains to the selected destination domains. Then select the domain type in the **Level** list and select the check boxes for the domains in the list where you want the projection coupling variable to exist. You can also select the domains directly in the drawing area in COMSOL Multiphysics and then select the **Use selected boundaries as destination** check box (or corresponding check box for other domain types). An asterisk in front of the name of a geometry or domain type (level) indicates that the projection coupling variable is available on that geometry and for that domain type.

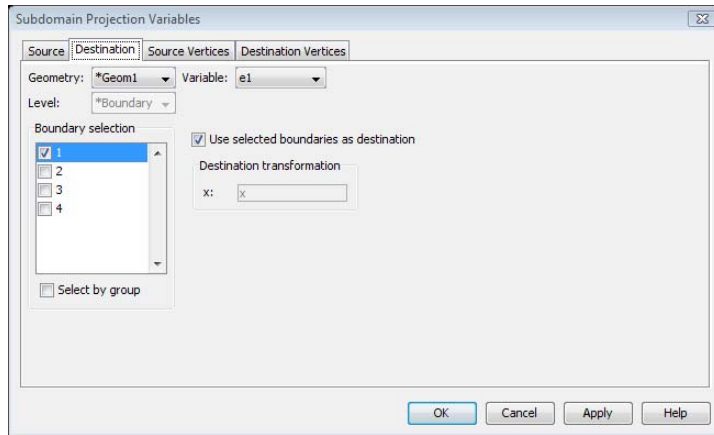


Figure 4-59: Specifying the destinations for the subdomain projection coupling variables (linear transformation).

You can change the variable in the **Variable** list. The default variable is the one selected on the source page.

Now specify the linear transformation by giving a set of points in the source geometry and in the destination geometry. The order of the vertices is significant. COMSOL Multiphysics constructs a linear transformation from the source domain to the destination domain using the subspaces spanned by the vertices. Denote the transformation rank by n , denote the source vertices by x_0, x_1, \dots, x_n , and denote the destination vertices by x'_0, x'_1, \dots, x'_n . After padding the source and destination vertices' vectors with zeros as necessary, the software solves the following matrix equation for a *transformation matrix* T and a *translation vector* V :

$$\begin{aligned} x'_0 &= Tx_0 + V \\ x'_1 - x'_0 &= T(x_1 - x_0) \\ &\dots \\ x'_n - x'_0 &= T(x_n - x_0) \end{aligned}$$

For the projection coupling variable you must have one more vertex in the source geometry than in the destination geometry. Choose n such that the source domain can be embedded in a subspace of dimension $n - 1$. For example, when the source domain is a straight line, it suffices to have two vertices; when it is a curve that can be

embedded in a plane, you need three vertices. Four vertices are necessary for a source domain that is a curved surface or a curve that can only be embedded in 3D.

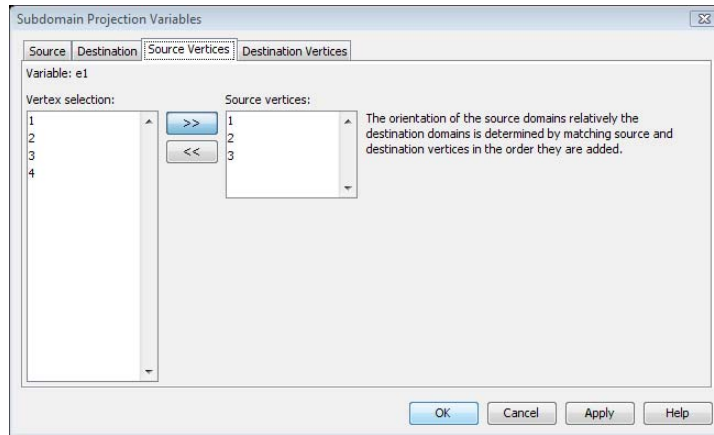


Figure 4-60: Specifying the source vertices for the subdomain projection coupling variables.

GENERAL TRANSFORMATION

The general transformation works similarly for extrusion and projection coupling variables. For details, see the discussion in the section “General Transformation” on page 277.

Choose the geometry of the source domain by opening the dialog box from that particular geometry. To define a projection coupling variable with general transformation, go to the **Options** menu and select **Projection Coupling Variables**. Then select **Subdomain Variables**, **Boundary Variables**, or **Edge Variables** depending on the intended source for the coupling variable.

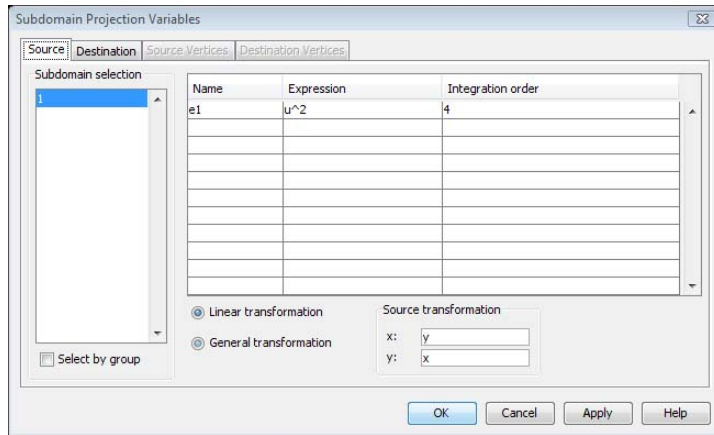


Figure 4-61: Specifying the sources for the subdomain projection coupling variables (general transformation).

In the dialog box, select source domains for the coupling variable in the domain selection list; in the example above it is the **Subdomain selection**. Make an entry for the coupling variable in the **Name** edit field, and then provide an expression for the selected source domains in the **Expression** edit field. Also make an entry in the **Integration order** edit field. Make sure that you have selected the **General transformation** button.

Now define the source transformation that maps the source domain to the intermediate domain of the same dimension. The source transformation has the same number of fields as the dimension of the source domain. You can use expressions containing space coordinates in the source geometry when defining the transformation. Remember that the transformation must be approximately linear within each mesh element.

Now click the **Destination** tab. First select the geometry in the **Geometry** list. Then select the domain type in the **Level** list, and select the check boxes for the domains in the list where you want the projection coupling variable to exist. You can also select the domains directly in the drawing area in COMSOL Multiphysics and then select the **Use selected boundaries as destination** check box (or the corresponding check box for other domain types). The expression on the source domains is available on the selected destination domains. An asterisk in front of the name of a geometry or domain type (level) indicates that the extrusion coupling variable is available on that geometry and for that domain type.

You can change the variable in the **Variable** list; the default variable is the one selected on the source page.

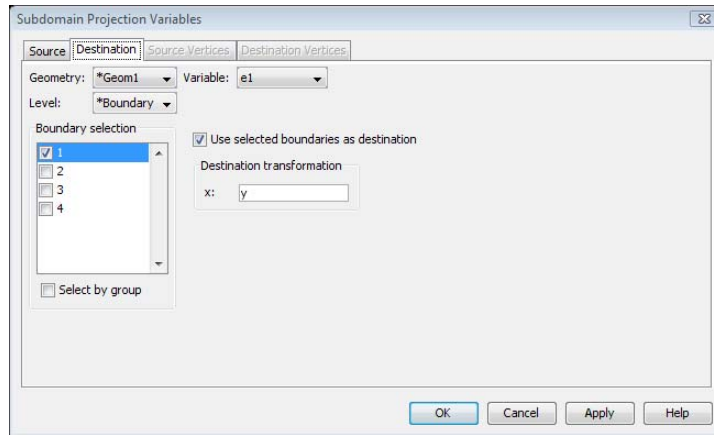


Figure 4-62: Specifying the destinations for the subdomain projection coupling variables (general transformation).

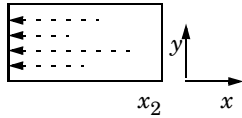
Now specify the destination transformation, which maps coordinates in the destination domain to the intermediate mesh. The destination transformation has one field less than the source transformation. When defining the transformation it is permissible to use expressions containing space coordinates in the destination geometry. You can also specify an arbitrary transformation, which can be highly nonlinear or noninvertible.

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

Projection Coupling Variables—Example

Consider the case of a single rectangular domain with Poisson’s equation. Integrate the solution squared along lines parallel to the x -axis and make the result available for postprocessing on the left boundary.

$$\begin{aligned} -\Delta u &= 1 && \text{on } \Omega \\ u &= 0 && \text{on } \partial\Omega \end{aligned}$$



The figure to the left illustrates the projection process. Project the integral of the solution squared on the boundary. The source transformation is y , and the destination transformation is x .

MODEL NAVIGATOR

- 1 In the **Model Navigator** select **2D** in the **Space dimension** list.
- 2 In the **PDE** folder select **PDE, Coefficient Form**. Make sure **Lagrange - Quadratic** is selected in the **Element** list.
- 3 Click **OK**.

GEOMETRY MODELING

- 1 Draw a square with sides of length 1 by clicking on the **Rectangle/Square** toolbar button, right-clicking on the origin, and then dragging to $(1, 1)$.
- 2 Click the **Zoom Extents** toolbar button.

PHYSICS SETTINGS

Boundary Conditions

Keep the default setting, that is, homogeneous Dirichlet boundary conditions.

Subdomain Settings

Keep the default setting, that is, Poisson's equation with a unit source.

Coupling Variables

- 1 In the **Options** menu, point to **Projection Coupling Variables**, and then click **Subdomain Variables** to open up the like-named dialog box.
- 2 Select Subdomain 1; enter the variable name $e1$ and the expression u^2 .
- 3 Click the **General transformation** button. Type y as the source transformation in the **x** edit field and then type x in the **y** edit field.
- 4 Click the **Destination** tab.
- 5 Select **Boundary** in the **Level** list and then select Boundary 1.
- 6 Type y as the destination transformation in the **x** edit field.
- 7 Click **OK**.

Computing the Solution

Click the **Solve** button on the Main toolbar.

Postprocessing and Visualization

- 1 In the **Plot Parameters** dialog box, click the **General** tab and clear the **Surface** check box.
- 2 Click the **Boundary** tab and then select the **Boundary plot** check box.
- 3 Type $e1$ in the **Expression** edit field on the **Boundary Data** tab.
- 4 Click the **Height Data** tab, select the **Height data** check box, and then type $e1$ in the **Expression** edit field.
- 5 Click **OK**.

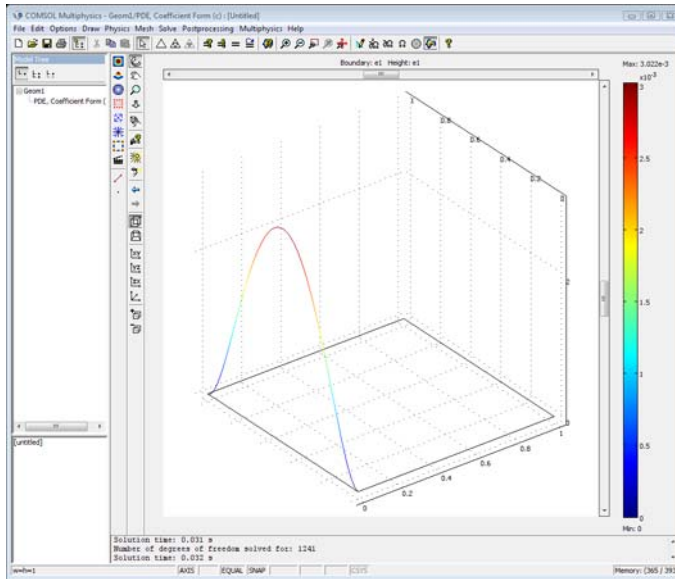
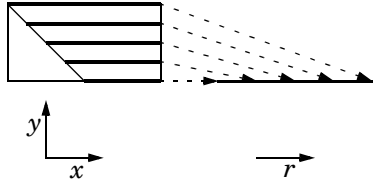


Figure 4-63: The integral of the solution squared on the left boundary.

The plot in Figure 4-63 shows the value of the integral of the solution squared on the left boundary, where the projection coupling variable $e1$ makes it available.

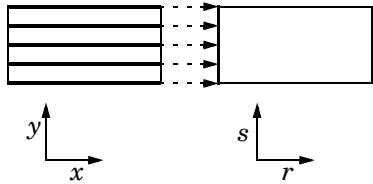
BRIEF EXAMPLES OF PROJECTION COUPLINGS



For each point r , the coupling variable returns the integral

$$v(r) = \int_{\substack{y = r/2 \\ (x, y) \in S_2}} u(x, y) dx$$

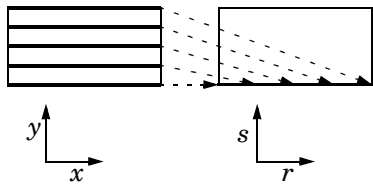
The source transformation mapping is y, x , and the destination transformation mapping is $r/2$.



For each point $(0, s)$, the coupling variable returns the integral

$$v(0, s) = \int_{\substack{y = s \\ (x, y) \in S_2}} u(x, y) dx$$

The source transformation mapping is y, x , and the destination transformation mapping is s .



For each point $(r, 0)$, the coupling variable returns the integral

$$v(r, 0) = \int_{\substack{y = r/2 \\ (x, y) \in S_2}} u(x, y) dx$$

The source transformation mapping is y, x , and the destination transformation mapping is $r/2$.

Note: The integration can also sweep nonrectangular domains. The integrals include only the source domains; they exclude other domains and the external area.

CHOOSING SOLUTION FORM

The definitions of these coupling variables usually includes a dependency on the solution. If the variables are intended for postprocessing tasks, this condition poses no problem, but if they are to be involved in the coefficients of the PDE or boundary conditions, they should make contributions to the Jacobian matrix. For this to be possible, the model must be solved using the *weak* form, which is the default solution form. (From the command line, you can use `flform` to convert the model to the weak form before solving.)

NONLOCAL COUPLINGS AND THE SPARSITY OF THE JACOBIAN

The Jacobian for problems formulated using the finite element method is usually large but sparse. This is because the solution at each mesh node can depend at most on the degrees of freedom at the neighboring mesh elements. However, by introducing coupling variables, you create nonlocal dependencies that fill up the rows and columns of the affected source and destination nodes. These additional elements might make the Jacobian matrix only slightly less sparse, with marginal effects on solution speed; it can also make it a great deal less sparse, in which case memory use and CPU time involved in solving the problem increases considerably. For this reason, take particular care when introducing nonlocal couplings.

Using Boundary Distance Variables

The boundary distance variables have the ability to make the distance to the closest boundary available on subdomains. These variables are primarily devised to be part of turbulence models, but you can use them for other purposes as well.

To define a boundary distance variable, go to the **Options** menu and choose **Boundary Distance Variables**.

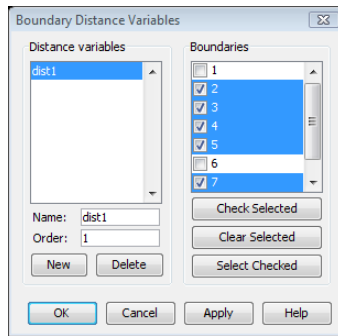


Figure 4-64: Specifying boundary distance variables.

Click **New** to create a new variable. Doing so generates a unique variable name. You can change the name afterward by editing the **Name** edit field. The **Order** edit field specifies the Lagrange order to use when computing the distances.

The check boxes beside the boundary numbers indicate which boundaries are included when computing the distance.

Click the **Check Selected** button below the list to select the check boxes of the boundaries highlighted in the list. This is equivalent to selecting the individual check boxes and is a quick way to select multiple check boxes. Click the **Clear Selected** button to clear the check boxes of the selected boundaries.

Use the **Select Checked** button to select the checked boundaries in the drawing area of the COMSOL Multiphysics user interface and in the selection lists in the dialog boxes for settings on boundaries, such as the **Boundary Settings** dialog box.

Note that COMSOL Multiphysics also defines these variables on the spatial frame for ALE applications (moving meshes).

Plotting Distances to Walls in Backstep Model

As an example of using a distance variable, plot the distances to the walls in the model of fluid flow over a backward-facing step. To do so, follow these steps:

MODEL NAVIGATOR

- 1 From the **File** menu, choose **Open Model Library**.
- 2 Select **Model Library>COMSOL Multiphysics>Fluid Dynamics>backstep**.
- 3 Click **OK**.

OPTIONS AND SETTINGS

Define the boundary distance variable for the distance to the wall boundaries.

- 1 From the **Options** menu, choose **Boundary Distance Variables**.
- 2 Click **New** to define a new boundary distance variable with the name **dist1**. Leave the order at the default values of 2.
- 3 Click the **Clear Selected** button to clear the default selection of all boundaries.
- 4 Select all the wall boundaries (Boundaries 2–5 and 7), either by clicking directly on the boundaries in the drawing area or, in the **Boundary Settings** dialog box, by selecting the **Select by group** check box and then selecting one of the wall boundaries.
- 5 In the **Boundary Distance Variables** dialog box, click the **Check Selected** button (see Figure 4-64 on page 295).
- 6 Click **OK**.

POSTPROCESSING AND VISUALIZATION

- 1 From the **Solve** menu, choose **Update Model** to evaluate the boundary distance function.
- 2 Click the **Plot Parameters** button on the Main toolbar.
- 3 In the **Plot Parameters** dialog box, click the **Surface** tab.
- 4 Type **dist1** in the **Expression** edit field.
- 5 Click the **General** tab and clear the **Streamline** and **Geometry edges** check boxes.
- 6 Click **OK** to plot the distance to the wall as a surface plot (see Figure 4-65 on page 297).

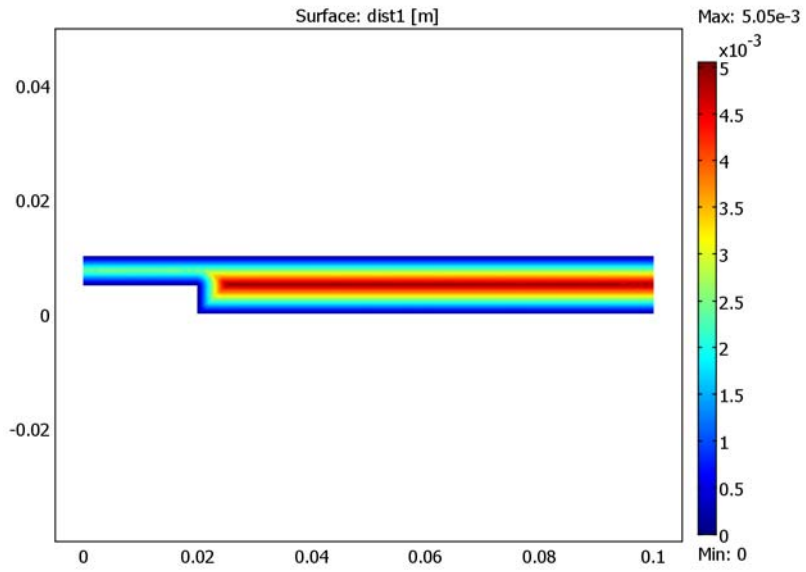


Figure 4-65: Visualization of the distance to the walls in the Backstep model.

Meshing

This chapter covers how to create and control the mesh for 1D, 2D, and 3D geometries in COMSOL Multiphysics. It also explains the possibilities for importing and exporting meshes in different formats. For additional information about meshing assemblies, see “Meshing Assemblies” on page 427 of the *COMSOL Multiphysics Modeling Guide*.

Creating Meshes

A mesh is a partition of the geometry model into small units of simple shapes. It is possible to create *free*, *mapped*, *extruded*, *revolved*, *swept*, and *boundary layer meshes*. The mesh generator creating free meshes, also referred to as the *free mesher*, is the only mesher that can be used on all types of geometry objects.

Mesh Elements

For a 1D geometry the mesh generator partitions the subdomains (intervals) into smaller intervals (or *mesh elements*). The endpoints of the mesh elements are called *mesh vertices*.

For a 2D geometry the mesh generator partitions the subdomains into *triangular* or *quadrilateral mesh elements*. If the boundary is curved, these elements represent only an approximation of the original geometry. The sides of the triangles and quadrilaterals are called *mesh edges*, and their corners are *mesh vertices*. A mesh edge must not contain mesh vertices in its interior. Similarly, the boundaries defined in the geometry are partitioned (approximately) into mesh edges, referred to as *boundary elements*, which must conform with the mesh elements of the adjacent subdomains. If there are isolated points in the geometry, these also become mesh vertices.

Similarly, in 3D the mesh generator partitions the subdomains into *tetrahedral*, *hexahedral*, or *prism* mesh elements whose faces, edges, and corners are called *mesh faces*, *mesh edges*, and *mesh vertices*, respectively. The boundaries in the geometry are partitioned into triangular or quadrilateral boundary elements. The geometry edges are partitioned into *edge elements*. Isolated geometry vertices become mesh vertices.

Meshing Techniques

For a 2D geometry, you can choose between creating a free mesh consisting of triangular elements, a free mesh consisting of quadrilateral elements, or a mapped mesh consisting of quadrilateral elements. Compared to a free mesh, a mapped mesh is structured in its pattern. If you want a mapped quadrilateral mesh on a geometry, you must build the geometry such that the subdomains are fairly regular in shape and do not contain holes.

For a 3D geometry you can choose between creating a free mesh containing tetrahedral elements or a swept mesh containing prism elements or hexahedral

elements. A swept mesh is structured in the sweep direction and can be either structured or unstructured orthogonally to the sweep direction.

You can also create a 3D mesh by extruding or revolving a 2D mesh. An extruded or revolved mesh is structured in the direction of the extrusion or revolution. When you extrude or revolve a 2D mesh into a 3D mesh, triangular elements and quadrilateral elements in the 2D mesh are extruded or revolved into prism (wedge) elements and hexahedral (brick) elements, respectively, in the resulting 3D mesh.

Furthermore, you can create a boundary layer mesh by inserting structured layers of elements along specific boundaries into an existing mesh.

Note: Different platforms have small differences handling floating-point operations. In some cases this may result in slightly different meshes on two computers for an identical model file.

Creating Free Meshes

The free mesher is available in all space dimensions, and you can use it for all types of geometries regardless of the topology and shape of a geometry. The free mesher automatically creates an unstructured mesh if you click the **Initialize Mesh** button on the Main toolbar and if you enter the Mesh mode with an empty mesh.

When you create a free mesh, the number of mesh elements is determined from the shape of the geometry and various mesh parameters. You control the mesh parameters for the free mesher by going to the **Mesh** menu and choosing **Free Mesh Parameters**, which opens the **Free Mesh Parameters** dialog box. With the mesh parameters you can specify local mesh-element sizes and control the element distribution. All mesh parameters aim at prescribing the maximum allowed mesh element size. This means that the size of a mesh element must not be larger than the minimum prescribed local element size determined by all mesh parameters.

CREATING FREE MESHES IN 3D

The **Free Mesh Parameters** dialog box for 3D geometries contains tabs named **Global**, **Subdomain**, **Boundary**, **Edge**, **Point**, and **Advanced**.

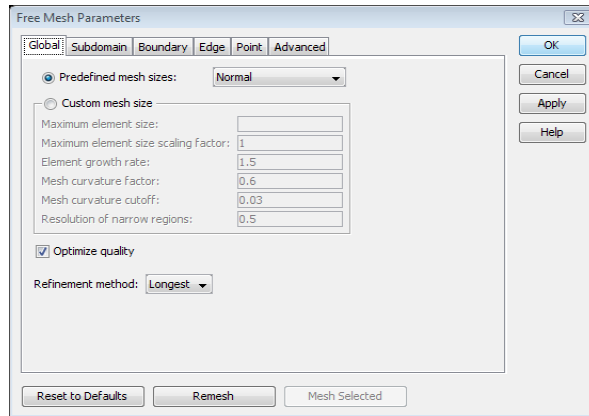
Click the **Remesh** button to delete the current mesh and create a new free mesh on the entire geometry according to the parameters in the dialog box. Clicking the **Reset to**

defaults button resets all changes of the parameters on all tabbed pages to their default values.

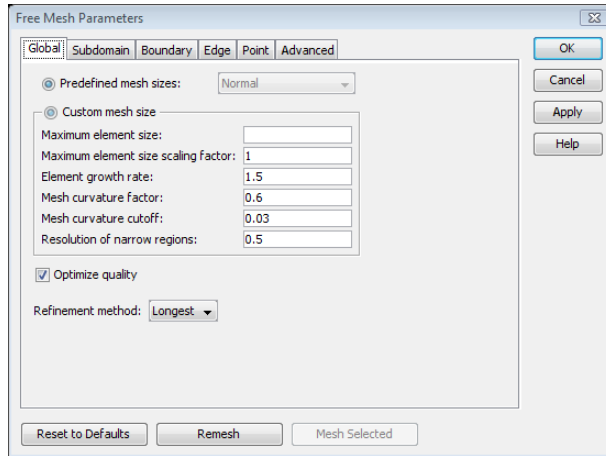
If you click the **Mesh Selected** button, the mesh generator meshes only the selected domains. You can read more about this in “Interactive Meshing” on page 346.

Global Parameters

The **Global** page allows you to set mesh parameters that determine element size and element distribution globally.



- In the **Predefined mesh sizes** list you can select a predefined mesh size that automatically determines the parameters in the **Custom mesh size** area. A predefined mesh size can generate an extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, or extremely coarse mesh. An indication on the Status bar shows the current predefined mesh size.
- Click the **Custom mesh size** button if you want to manually specify the global mesh parameters.

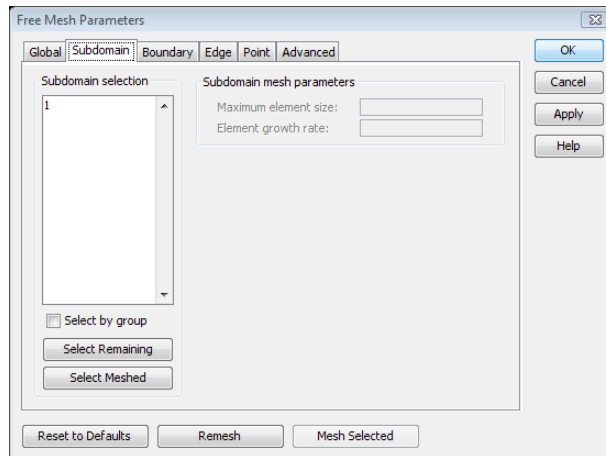


- The value in the **Maximum element size** edit field specifies the maximum allowed element size, which by default is 1/10th of the maximum distance in the geometry. This parameter is optional.
- The **Maximum element size scaling factor** comes into play if you do not explicitly give a maximum element size. In that case the mesh generator multiplies the default maximum element size by this factor; its default value is 1.
- The **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. The default value is 1.5, that is, element size can grow by 50% (approximately) from one element to another.
- The value in the **Mesh curvature factor** edit field determines the size of boundary elements compared to the curvature of the geometric boundary. The curvature radius multiplied by the curvature factor, 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. The default value is 0.6.
- The parameter in the **Mesh curvature cutoff** edit field prevents the generation of many elements around small curved parts of the geometry. This parameter must be a positive value, and it has a default value of 0.03. When the radius of curvature is smaller than the product of the mesh curvature cutoff value times the maximum distance in the geometry, the mesh generator considers the radius of curvature as being this product.

- In the **Resolution of narrow regions** edit field you control the number of layers of elements that are created in narrow regions (approximately). The value must be a positive 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.
- The **Optimize quality** check box controls if the mesh generator carries out quality optimization at the end of the meshing process.
- From the **Refinement method** list you select whether the mesh generator uses the regular or longest refinement method when refining the mesh. For more information on these choices see “Refinement Methods” on page 344.

Subdomain and Point Parameters

On the **Subdomain** and **Point** pages you can specify mesh parameters per subdomain and vertex. Select domains in the domain selection lists or by clicking them in the drawing area.



- An entry in the **Maximum element size** edit field specifies that parameter for the selected domains.
- An entry in the **Element growth rate** edit field specifies that parameter for the selected domains.

The default values of the local parameters are identical to the defaults for the corresponding global parameters.

Selecting By Group

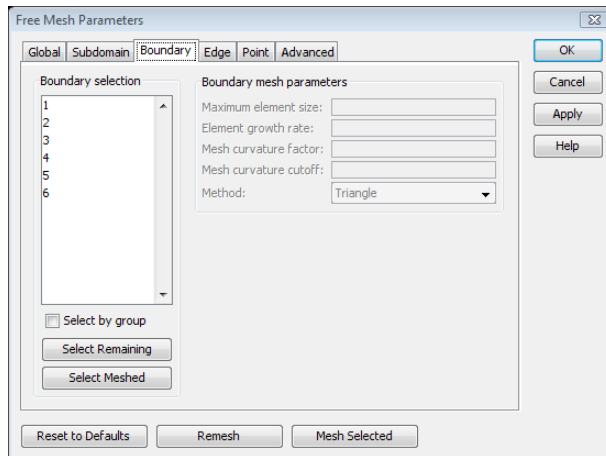
If you select the **Select by group** check box, selecting a domain automatically selects all domains with equal settings. This can be convenient when changing values that you have previously entered on a large set of domains.

Selecting the Remaining Domains or the Meshed Domains

Click the **Select Remaining** button to select all domains that have not yet been meshed, or click **Select Meshed** to select all meshed domains.

Boundary and Edge Parameters

On the **Boundary** and **Edge** pages you can specify mesh parameters per face and edge domain. In addition to the parameters available on the **Subdomain** and **Point** pages, it is possible to specify the resolution of curved faces and edges locally.

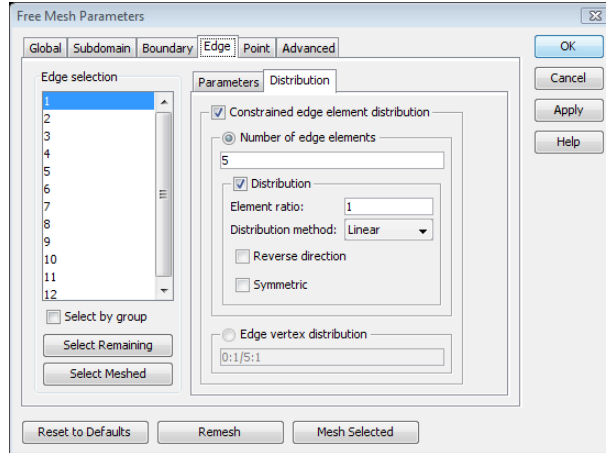


- An entry in the **Mesh curvature factor** edit field specifies that parameter for the selected domains.
- An entry in the **Mesh curvature cutoff** edit field specifies that parameter for the selected domains.

On the boundary page it is possible to specify the meshing method to be used for each boundary. To do this, select either **Triangle**, **Triangle (advancing front)**, or **Quad** from the **Method** list. The first and second choices result in a triangular mesh, and the third gives a quadrilateral mesh on the selected boundaries.

Constraining the Edge Mesh

On edges in 2D and 3D it is possible to explicitly specify the number of elements and even the distribution of elements. You do this by selecting the **Constrained edge element distribution** check box on the **Distribution** page.

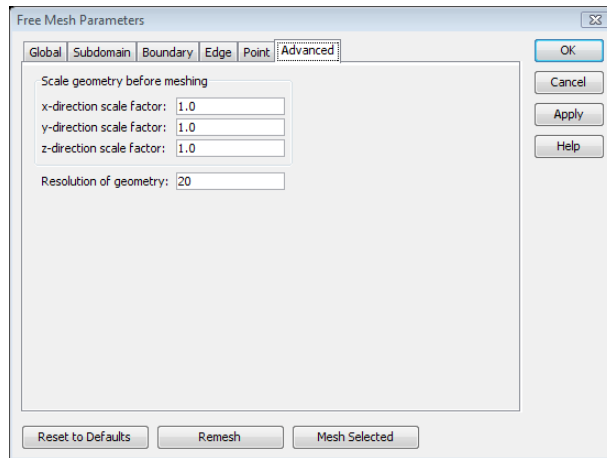


In the **Number of edge elements** area you specify the number of elements on the edge. If you clear the **Distribution** check box, the mesh generator distributes the edge elements along the edge according to the local element size determined by the ordinary free mesh parameters. If you select the check box, you specify the ratio in size between the last element and first element along the edge in the **Element ratio** edit field. Furthermore, in the **Distribution method** list you specify if the element distribution is linear or exponential along the edge. Select the **Reverse direction** check box to let the distribution be defined in the opposite edge direction and select the **Symmetric** check box to get a symmetric distribution.

Alternatively, click the **Edge vertex distribution** button to explicitly specify the distribution of mesh vertices along the edge. In the corresponding edit field you specify a vector-valued expression of strictly increasing values starting with zero, specifying the relative arc length values of the mesh vertices along the edge. For more information on vector-valued expressions see “Entering Vector-Valued Expressions” on page 145.

Advanced Parameters

The **Advanced** page sets parameters controlling the geometry scaling factors and the resolution of the faces.



- In the **Scale geometry before meshing** area you can specify the factors by which COMSOL Multiphysics 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. For more information on this topic see “Meshing Thin Structures” on page 342.
- The parameter in the **Resolution of geometry** edit field controls the number of points placed in each surface parameter direction on each face to resolve the geometry. For extremely curved geometries you might need to increase this value. The value must be a positive integer. The default value is 20.

CREATING FREE MESHES IN 1D AND 2D

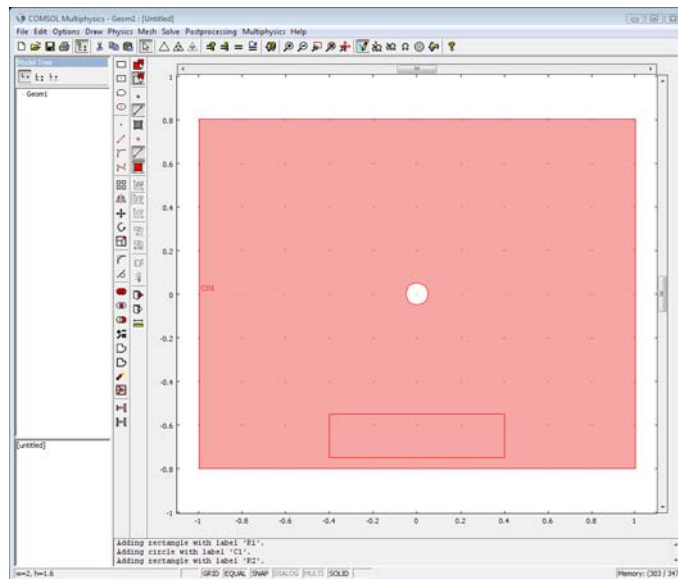
The **Free Mesh Parameters** dialog box in 2D contains the same set of parameters as in 3D although some of the parameters have different default values. On the **Subdomain** page in 2D it is possible to specify the meshing method for each subdomain to either **Triangle** or **Quad**, in the same way as for boundaries in 3D. In 1D only a subset of the parameters in 3D are available.

CONTROLLING THE MESH DENSITY OF A FREE MESH

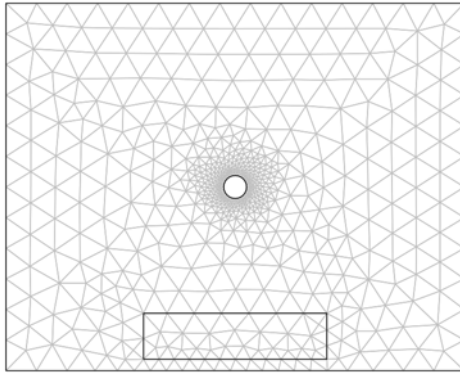
It is easy to create a free mesh by clicking the **Initialize Mesh** button. In most cases, doing so creates an appropriate mesh. However, in finite element analyses it is often necessary to resolve a specific region in more detail, for instance, a particular region of a geometry that features more complicated solution patterns. The local element sizes and the distribution of elements within the geometry is controlled via the parameters in the **Free Mesh Parameters** dialog box.

This section shows the effects of the different mesh parameters on a 2D geometry.

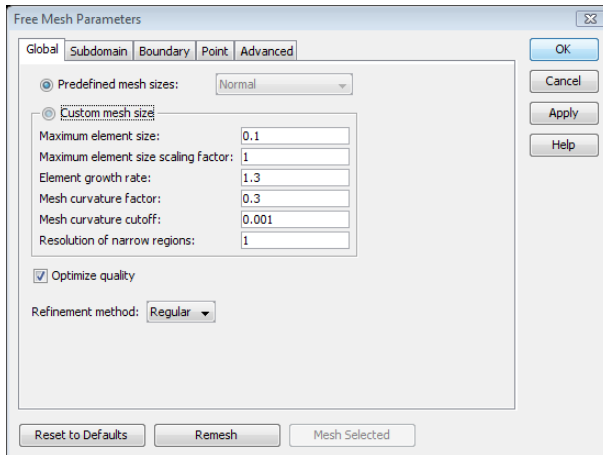
- 1 Start a new 2D model.
- 2 In Draw mode, create a rectangle with corners at $(-1, -0.8)$ and $(1, 0.8)$.
- 3 Create a circle with radius of 0.05 centered at the origin, for example, by choosing **Specify Objects>Circle** from the **Draw** menu.
- 4 Subtract the circle from the rectangle.
- 5 From the **Draw** menu choose **Specify Objects>Rectangle** and create a rectangle with the width equal to 0.8, height equal to 0.2, and with the lower left corner positioned at $(-0.4, -0.75)$.
- 6 Select all objects by pressing Ctrl+A, and then click the **Union** button on the Draw toolbar.



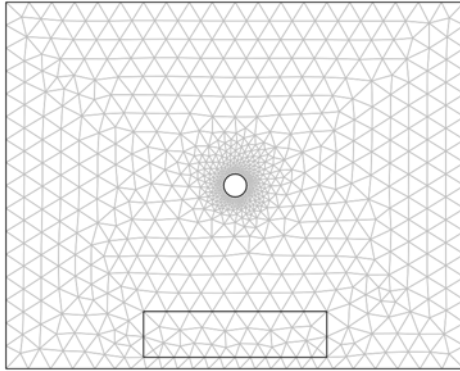
- 7 Create a mesh using the default mesh settings by clicking the **Initialize Mesh** button on the Main toolbar.



- 8 Open the **Free Mesh Parameters** dialog box by choosing **Free Mesh Parameters** in the **Mesh** menu, then click the **Global** tab.
- 9 By default, the maximum element size used is 1/15th (in 2D) of the maximum distance in the geometry. In this case the default value of the maximum element size is 2/15. However, you can explicitly specify the maximum element size in the **Maximum element size** edit field in the **Custom mesh size** area. Create a finer mesh by selecting **Custom mesh size** and typing 0.1 in the **Maximum element size** edit field.

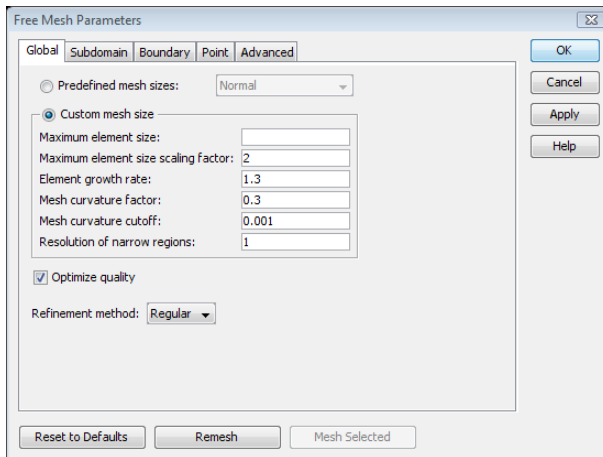


10 Click the **Remesh** button on the Main toolbar.

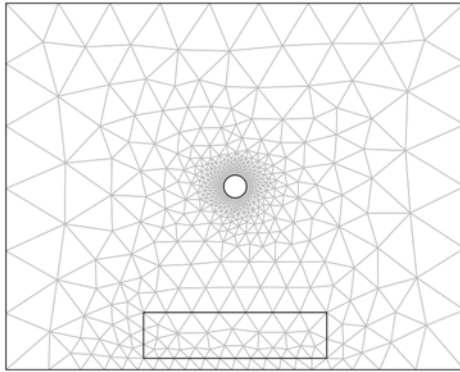


Another way of controlling the maximum element size globally without having to bother about the scale of the geometry is to use the maximum element size scaling factor. If the maximum element size is not given, the maximum element size is multiplied by the value specified in the **Maximum element size scaling factor** edit field.

1 Create a coarser mesh than the default mesh by clearing the **Maximum element size** edit field and typing 2 in the **Maximum element size scaling factor** edit field.

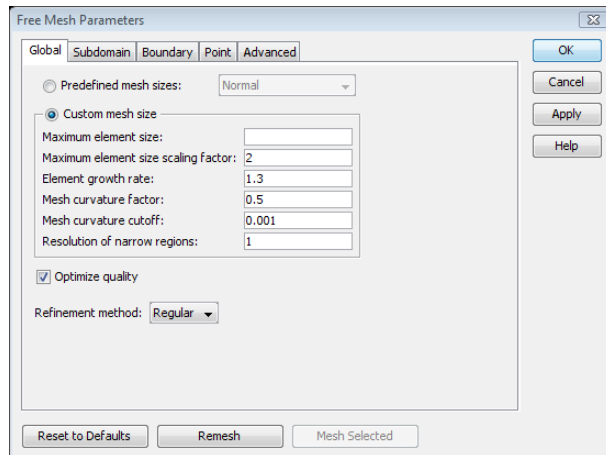


2 Click **Remesh**.

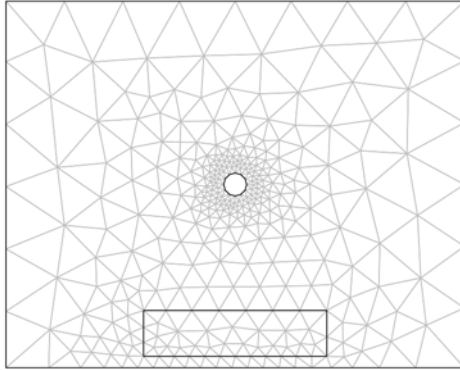


Notice that increasing the maximum element size does not influence the mesh density close to the circular hole. In this region, the mesh curvature factor is the parameter that provides the active constraint.

- 1 Create a new mesh that is coarser in the region close to the circular hole by increasing the value specified in the **Mesh curvature factor** edit field to 0.5.

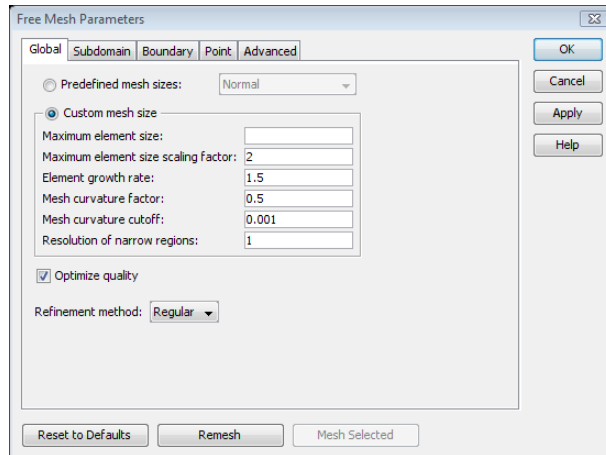


2 Click **Remesh**.

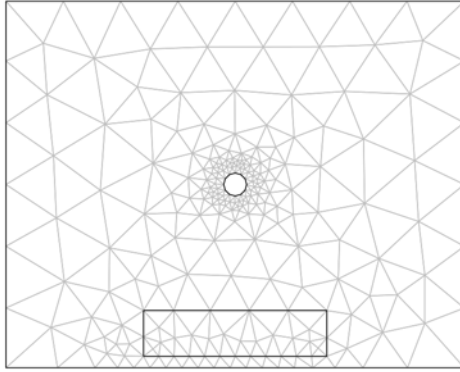


To decrease the number of elements even more, you can increase the element growth rate, which determines how fast the element size can grow from the circular hole.

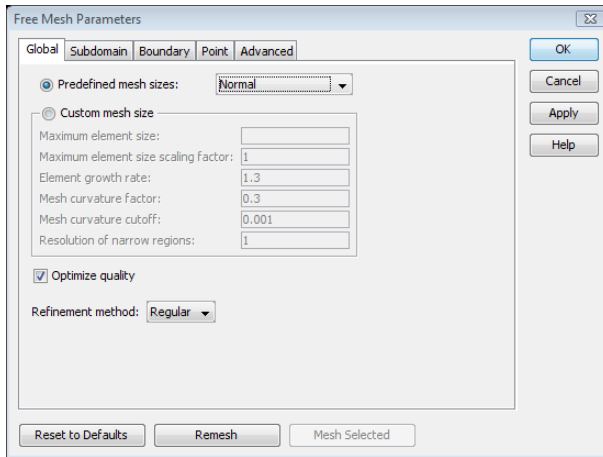
- 1 Create a mesh that is even coarser by increasing the value in the **Element growth rate** edit field to 1.5.



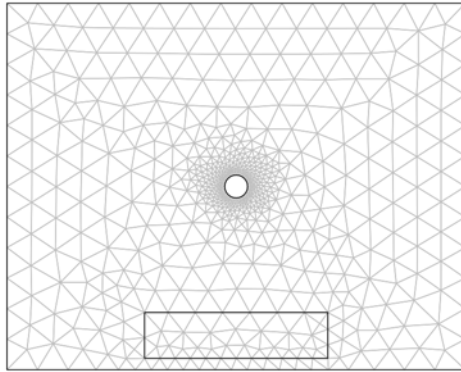
2 Click **Remesh**.



3 Create a default mesh by clicking the **Reset to Defaults** button.

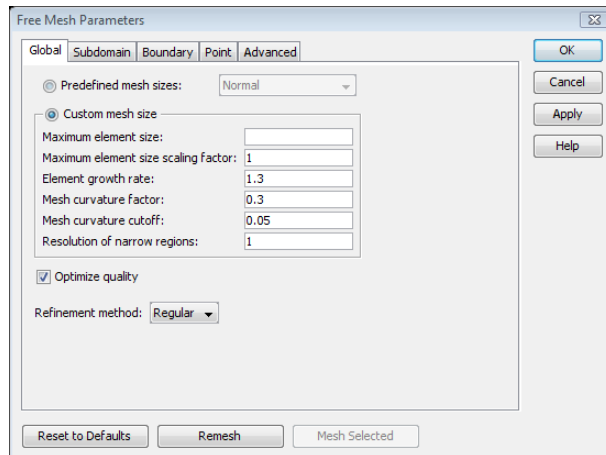


4 Click **Remesh**.

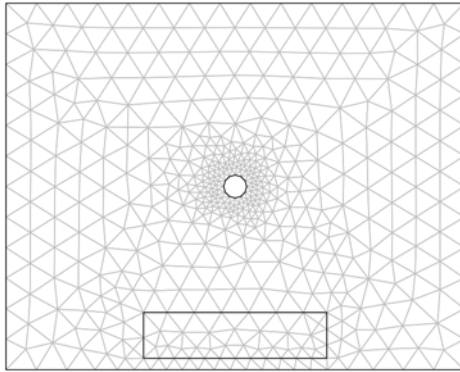


You can prevent the circular hole from getting a fine resolution by increasing the mesh curvature cutoff value. The hole's radius of curvature is considered to be the product of the mesh curvature cutoff value times the maximum distance in the geometry—if this product is greater than the radius of curvature. In this case the mesh curvature cutoff value needs to be greater than 0.025 to take effect.

- 1 Prevent the fine resolution of the circular hole by increasing the value in the **Mesh curvature cutoff** edit field to 0.05.

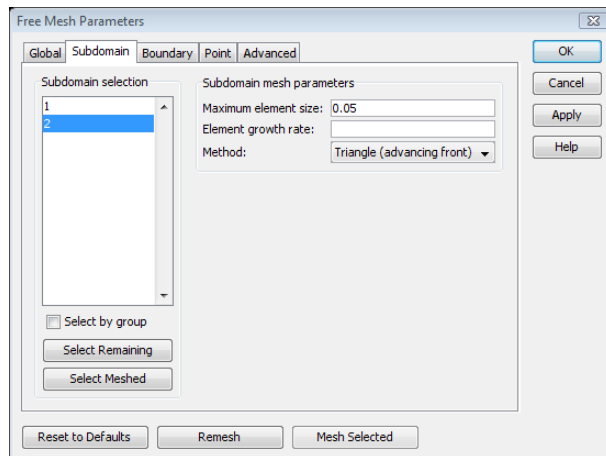


- 2 Click **Remesh**.

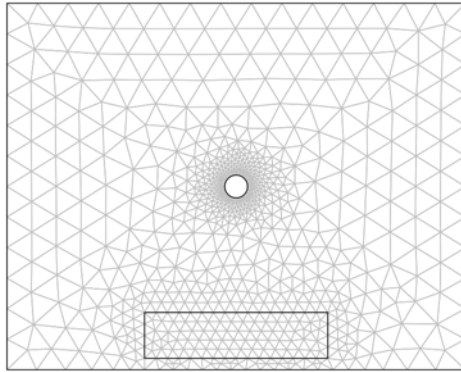


You can specify some mesh parameters locally. For example, in 2D, per subdomain, edge, or vertex. The value of a certain mesh parameter that the mesher uses locally is the minimum of the local value and global value of the corresponding parameter.

- 1 Reset the mesh settings by selecting **Normal** in the **Predefined mesh sizes** list.
- 2 Click the **Subdomain** tab.
- 3 Select Subdomain 2 in the **Subdomain selection** list and type 0.05 in the **Maximum element size** edit field.

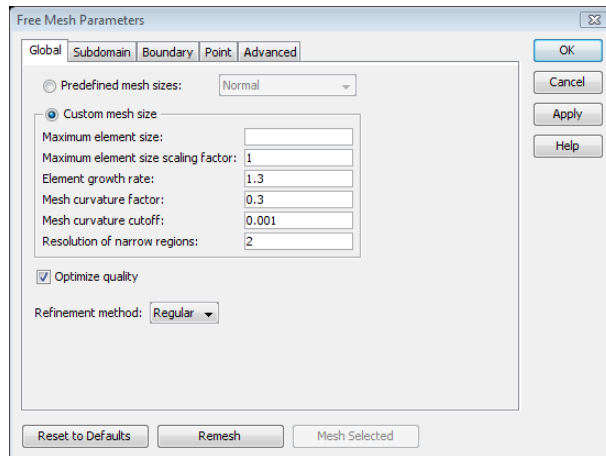


- 4 Click **Remesh** to create a new mesh that is finer in Subdomain 2, that is, the interior subdomain.

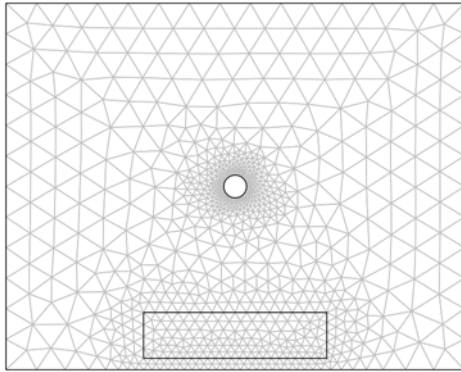


By default the mesh generator adjusts the boundary mesh along boundaries adjacent to a narrow region such that there is approximately one layer of elements, which are isotropic in size, in the narrow region. To control the number of layers of elements, use the **Resolution of narrow regions** edit field.

- 1 Click the **Global** tab.
- 2 Select **Custom mesh size** and type 2 in the **Resolution of narrow regions** edit field.



- 3 Click **Remesh** to create a new mesh with approximately two layers of elements in the narrow region along the lower horizontal boundary.



FREE MESHING METHODS ON FACES

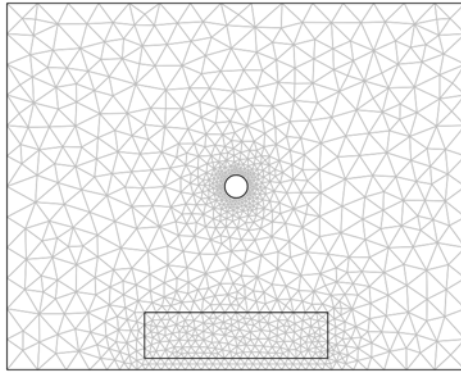
Three different methods are available for creating meshes on subdomains in 2D and boundaries in 3D. You can choose to create a triangular mesh using either a Delaunay based method, an advancing front based method, or to create a quadrilateral mesh. On subdomain in 2D the free mesher uses the method based on the advancing front algorithm as default while it uses the method based on the Delaunay algorithm as default on boundaries in 3D.

Using the Delaunay Method in 2D

Start with the geometry and mesh settings from the last example of the previous section. First open the **Free Mesh Parameters** dialog box, then follow these steps:

- 1 Click the **Subdomain** tab.
- 2 Select subdomains 1 and 2 from the **Subdomain selection** list.
- 3 Choose **Triangle** from the **Method** list.

4 Click **Remesh** to create a new mesh using the Delaunay method.



Free Quadrilateral (Quad) Meshes

It is also possible to create a free quad mesh on subdomains in 2D and on boundaries in 3D. For example, to create a quad mesh in 2D, open the **Free Mesh Parameters** dialog box by choosing **Free Mesh Parameters** in the **Mesh** menu, then click the **Subdomain** tab. Choose **Quad** from the **Method** list.

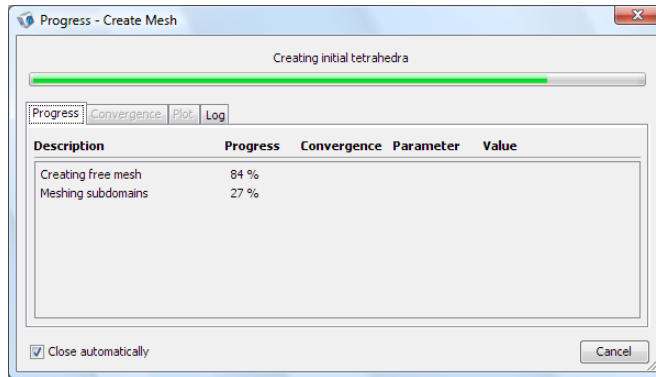
A necessary condition for a quadrilateral mesh is that each subdomain meshed with quads must be surrounded by an even number of edge elements. If you do not specify a constrained edge element distribution, the mesh generator automatically tries to distribute the edge elements so that this condition is met. If you choose to explicitly specify the edge element distribution, you must make sure to fulfill this condition. For more information about controlling the edge element distribution, see “Constraining the Edge Mesh” on page 306.

The generated quad mesh is not guaranteed to contain only quadrilateral elements. If the algorithm for some reason fails to mesh the entire domain with quad elements, or if the quality of a quad element is very low, the mesh generator uses some triangular elements instead.

Meshing Progress Indication

When you create a mesh, a **Progress** window opens, and it shows a message and a progress bar. The progress bar updates during mesh generation, and the message

describes the current stage of the mesh generation, making it possible to estimate the time remaining.



Canceling the Mesh Generation

If you realize that the geometry being meshed will produce a mesh with too many elements, or that it is just taking too long, click the **Cancel** button. The mesh generator then attempts to cancel the mesh generation as soon as possible, and COMSOL Multiphysics returns to where it was before starting to mesh.

Closing the Progress Window Automatically

By default the **Progress** window disappears when the mesh generation is completed. Clear the **Close automatically** check box to make the **Progress** window stay on top when the mesh generation is finished. This is useful when you want to have a closer look at the **Log** page. If the minimum element quality of the generated mesh is extremely poor ($< 10^{-4}$), the **Progress** window does not close, the **Log** tab becomes visible, and a warning message appears in red. For more information on mesh-element quality see “Visualizing the Mesh Quality” on page 371.

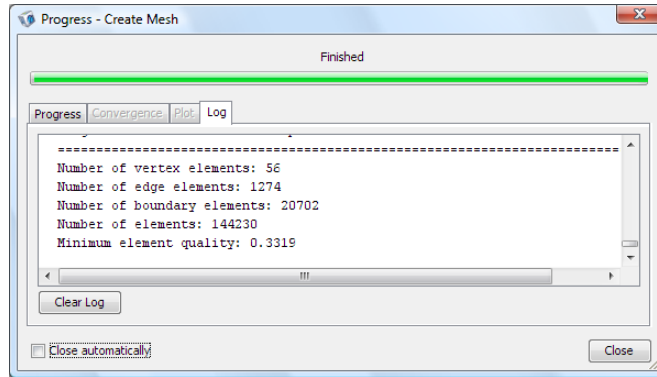
The Progress Page

On the **Progress** page, the value in the **Progress** column shows an estimate of the progress for the mesh generation. By the end of the mesh-generation process the **Value** fields display the current number of elements generated by the meshing operation.

The Log Page

The **Log** page displays the number of vertex elements, edge elements, boundary elements, and mesh elements as soon as these numbers are determined. It also prints

the minimum element quality when the mesh generation is finished. If the minimum element quality is extremely poor ($< 10^{-4}$), a warning appears.



The Convergence Page and the Plot Page

The **Convergence** and **Plot** pages are empty during mesh generation.

Creating Mapped Meshes in 2D

To create a mapped mesh consisting of quadrilateral (quad) elements, use the **Mapped Mesh Parameters** dialog box that you open from the **Mesh** menu. The quad mesh is generated using a mapping technique. For each subdomain, the mapping algorithm defines a regular grid on a logical unit square and then maps it onto the real geometry using transfinite interpolation.

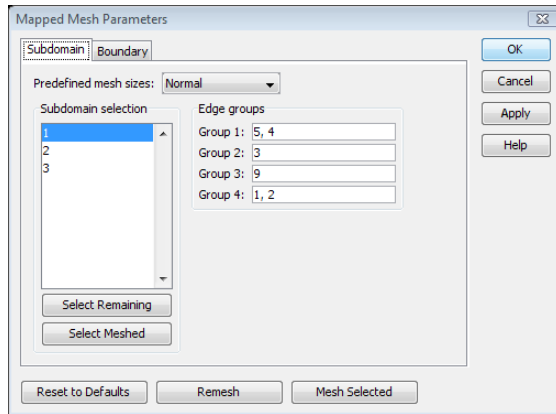
Click the **Remesh** button to delete the current mesh and create a mapped mesh on all subdomains according to the parameters in the dialog box. By clicking the **Reset to defaults** button, all changes of the parameters on all tabbed pages are reset to their default values.

If you click the **Mesh Selected** button, only the selected subdomains are meshed. You can read more on this in “Interactive Meshing” on page 346.

THE SUBDOMAIN PAGE

On the **Subdomain** page you specify the relationship between the boundary segments for each subdomain and the four sides of the logical unit square in the **Edge groups** area. In the **Predefined mesh sizes** list you can select a setting that determines the size of the

elements in the mesh. A predefined mesh size can generate an extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, or extremely coarse mesh.



Selecting the Remaining Domains or the Meshed Domains

Click the **Select Remaining** button to select all subdomains that have not yet been meshed, or click **Select Meshed** to select all meshed subdomains.

THE BOUNDARY PAGE

On the **Boundary** page you can control the number of elements and the distribution of elements for each edge. Due to the mapping technique, the number of elements for a subdomain is uniquely determined from the number of elements on the edges about the subdomain.

REQUIREMENTS ON THE GEOMETRY

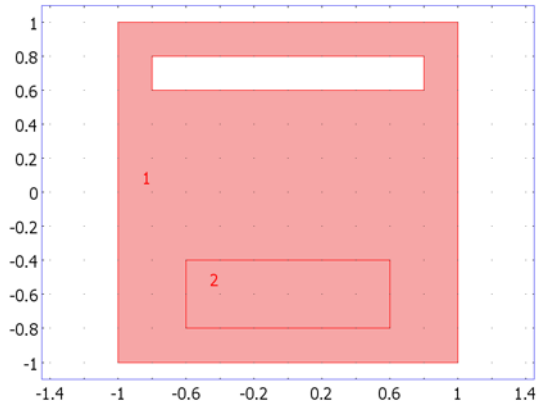
For the mapping technique to work properly, the geometry must be fairly regular. The following conditions must be satisfied:

- Each subdomain must be bounded by at least four boundary segments.
- Each subdomain must be bounded by only one connected boundary component (that is, no holes are allowed).
- The subdomains must not contain isolated vertices or isolated boundary segments.
- The shape of each subdomain must not differ too much from a rectangle.

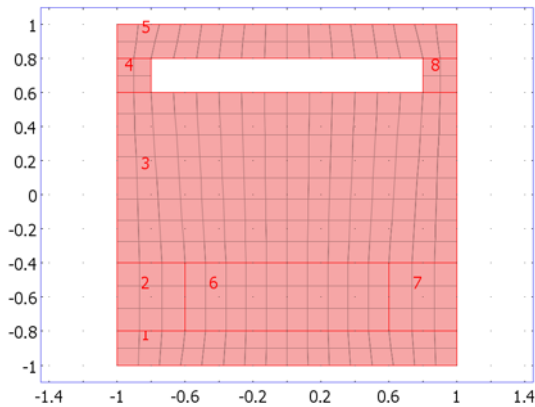
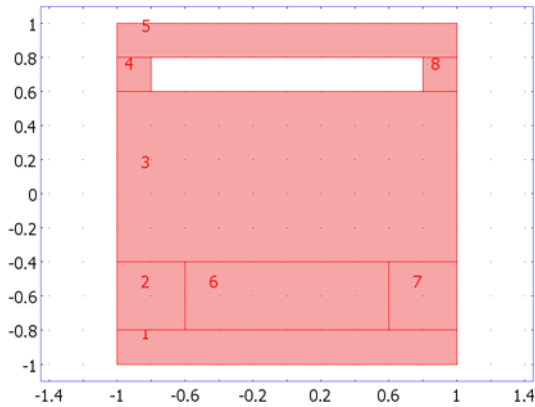
For a geometry model that initially does not meet these criteria, it is often possible to modify it such that a mapped mesh can be successfully generated.

Example—Creating a Mapped Mesh in 2D

The following geometry model does not meet the second condition because Subdomain 1 is bounded by three connected boundary components.



However, introducing some additional interior boundary segments in the geometry, which divide Subdomain 1 into several subdomains, makes it possible to create a mapped mesh.



THE MAPPING TECHNIQUE

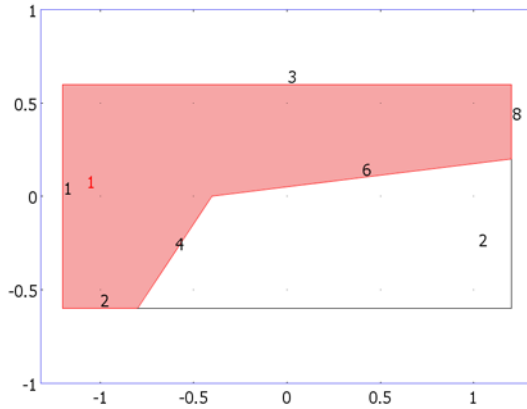
To create a quadrilateral mesh for each subdomain, the mapped mesher maps a regular grid defined on a logical unit square onto each subdomain. The mapping method is based on transfinite interpolation. The setting in the **Predefined mesh sizes** list on the **Subdomain** page and the settings on the **Boundary** page 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.

You specify the relationship between the four sides of the logical unit square and the boundary segments around each subdomain on the **Subdomain** page. If a subdomain is

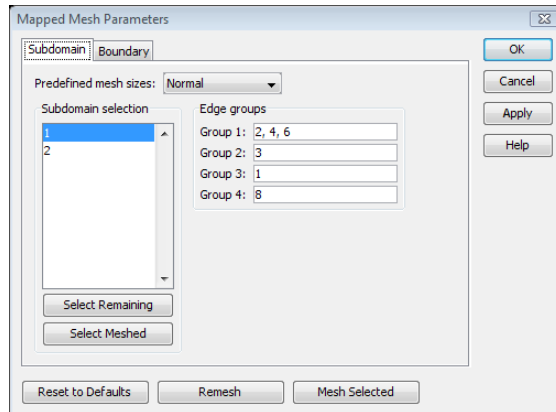
bounded by exactly four boundary segments, this relationship is trivial. Otherwise, the defaults determined by the mapping method are based on a criterion related to the sharpest angle between the boundary segments. However, in some cases you might want to modify these defaults.

Example—Mapped Meshes

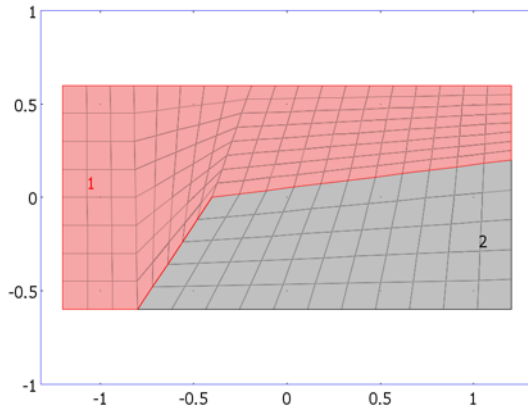
In this geometry model, Subdomain 1 is bounded by six boundary segments.



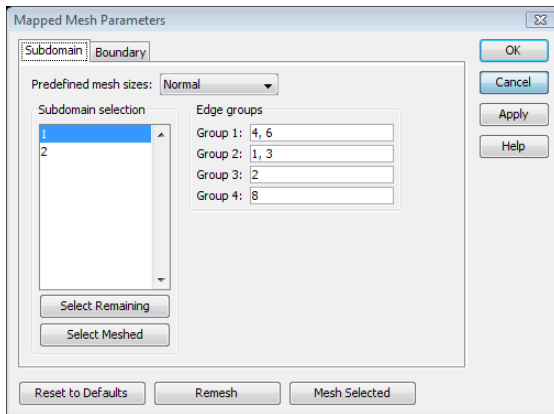
You can see the default grouping of the boundary segments for this subdomain by selecting Subdomain 1 in the **Subdomain selection** list. Each group in the **Edge groups** area corresponds to one side of the logical unit square.

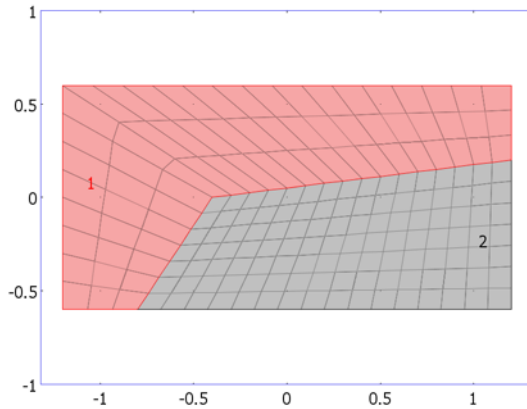


Using the default settings, the following mesh is generated.



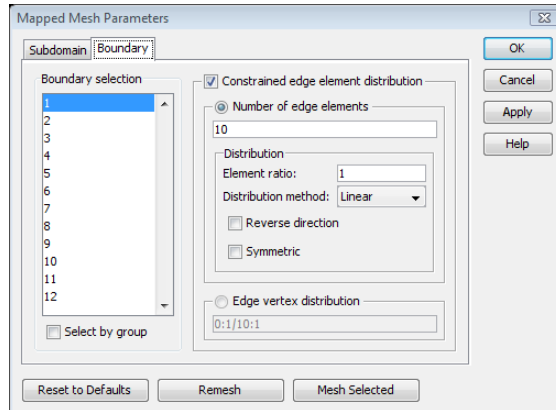
When you modify the edge groups, a different mesh is created.





CONTROLLING THE MESH DENSITY

The overall mesh density of a mapped mesh is uniquely determined from the distribution of edge elements on the boundary segments. You control the edge-element distribution on the **Boundary** page. If you want to force a specific edge-element distribution on a boundary segment, select the **Constrained edge element distribution** check box.



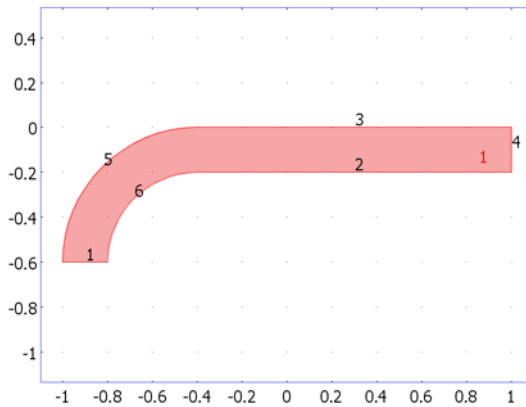
In the **Number of edge elements** area you specify the number of elements on the edge. You dictate the ratio in size between the last element and first element along the edge in the **Element ratio** edit field, and you specify whether the element distribution is linear or exponential in the **Distribution method** list. Select the **Reverse direction** check box to let the distribution be defined in the opposite edge direction and select the **Symmetric** check box to get a symmetric distribution.

Alternatively, by selecting the **Edge vertex distribution** option button you can explicitly specify the distribution of mesh vertices along the edge. In the corresponding edit field you give a vector-valued expression of strictly increasing values starting with zero, specifying the relative arc length values of the mesh vertices along the edge. For more information on vector-valued expression see “Entering Vector-Valued Expressions” on page 145.

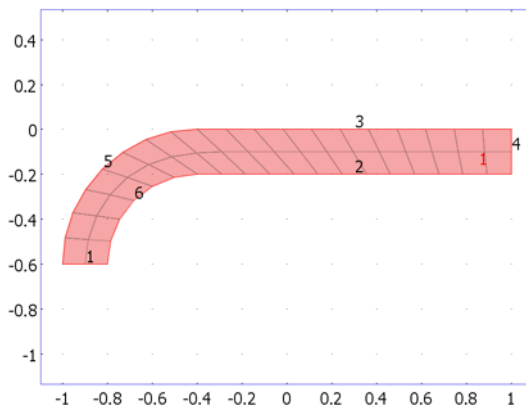
On unconstrained boundary segments, the mesh generator determines the number of edge elements according to the element size given with the setting in the **Predefined mesh sizes** list.

Example

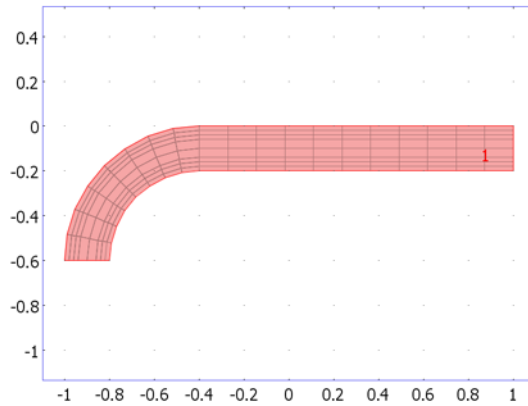
Study the edge labels for the following geometry.



Using the default settings, COMSOL Multiphysics creates this mapped mesh:



Constraining the number of edge elements to 8 for Boundary 6 and by typing the vector `0:0.1:0.3 0.5 0.7:0.1:1` in the **Edge vertex distribution** edit field for Boundaries 1 and 4 results in this anisotropic mapped mesh:



If the mapped quad mesher fails in determining matching edge discretizations of opposite sides in the logical mesh, an error message reports the corresponding subdomain index. Often you can avoid this error by constraining the number of edge elements on the corresponding boundary segments such that matching edge discretizations are defined.

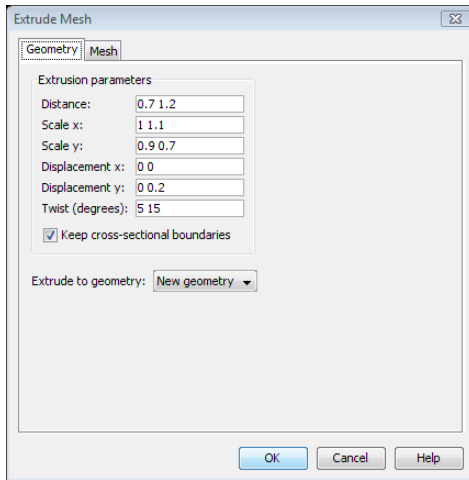
Extruding and Revolving 2D Meshes

It is also possible to create a 3D mesh by extruding or revolving a 2D mesh. Extrude or revolve a triangular mesh to create a prism (wedge) mesh, and extrude or revolve a quadrilateral mesh to create a hexahedral (block) mesh. The mesh generator creates the corresponding extruded or revolved geometry together with the mesh.

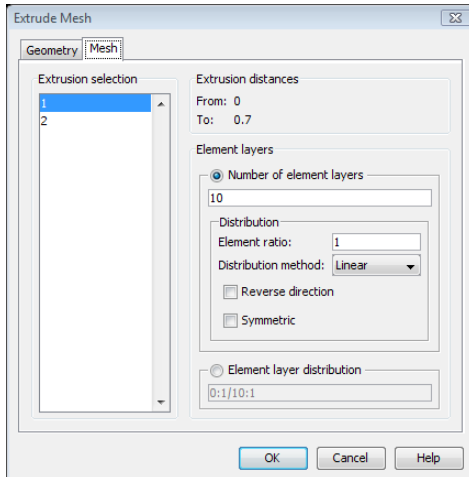
EXTRUDING A 2D MESH

Choosing **Extrude Mesh** from the **Mesh** menu opens the **Extrude Mesh** dialog box. On the **Geometry** page you specify the properties of the geometry extrusion. For more

information on geometry-extrusion properties see “Extruding a 2D Geometry” on page 64.



On the **Mesh** page you specify the properties for the extruded mesh. In the **Extrusion selection** list you select a section in the extruded geometry. You can see the corresponding extrusion distances in the **Extrusion distances** area.



Specifying the Element Layers

In the **Element layers** area you specify the element-layer distribution in the extrusion direction. In the **Number of element layers** area you specify the number of element layers along the extrusion. You dictate the ratio in size between the last element layer

and first element layer along the extrusion in the **Element ratio** edit field, and you specify whether the element layer distribution is linear or exponential in the **Distribution method** list. Select the **Reverse direction** check box to let the distribution be defined in the opposite extrusion direction and select the **Symmetric** check box to get a symmetric distribution.

Alternatively, by selecting the **Element layer distribution** option button you can explicitly dictate the distribution of element layers along the extrusion. In the corresponding edit field you specify a vector-valued expression (of strictly increasing values starting with zero) that specifies the relative arc length values of the element layers. For more information on vector-valued expression see “Entering Vector-Valued Expressions” on page 145.

The following figure shows the result of a mesh extrusion with hexahedral elements.

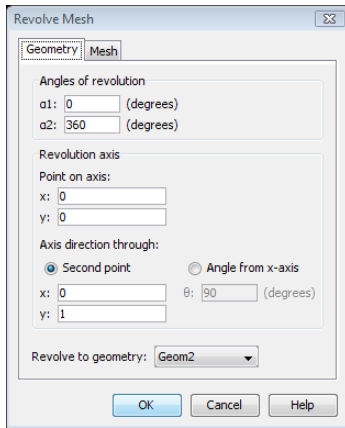


Extruded mesh with hexahedral elements

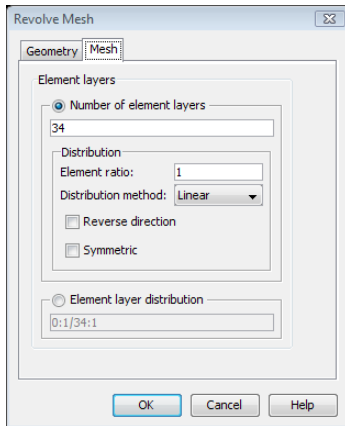
REVOLVING A 2D MESH

Choosing **Revolve Mesh** from the **Mesh** menu opens the **Revolve Mesh** dialog box. On the **Geometry** page you specify the properties of the geometry revolution. For more

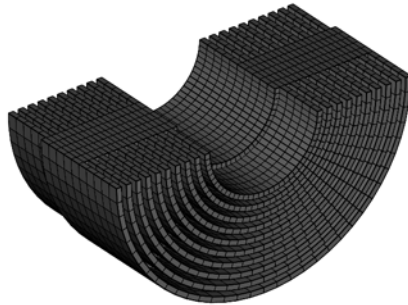
information on geometry-revolution properties see “Revolving a 2D Geometry” on page 65.



On the **Mesh** page you specify the element distribution in the **Element layers** area in the same way as when extruding a mesh (see “Extruding a 2D Mesh” on page 328). Note, however, that when you revolve a mesh, additional element layers might be introduced due to the division of the revolved geometry into sections.



The next figure shows the result of a mesh revolution with hexahedral elements.

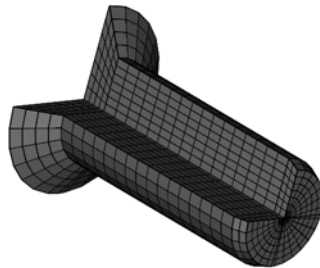


Revolved mesh with hexahedral elements

Singular Mesh Revolutions

A revolution is said to be singular if any part of the geometry touches the revolution axis. A singular revolution of a mesh is valid if the all elements adjacent to the revolution axis are quadrilateral elements with exactly one mesh edge on the revolution axis. The quadrilateral elements along the revolution axis are revolved into prism elements in the resulting 3D mesh.

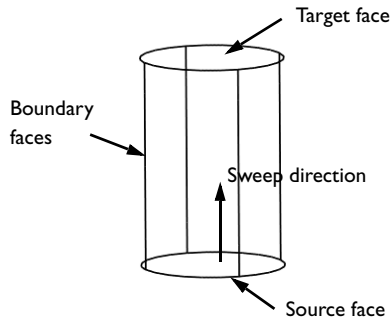
The following figure shows the result of a singular mesh revolution. The revolved mesh contains both prism and hexahedral elements.



Singularly revolved mesh consisting of prism elements and hexahedral elements

Creating Swept Meshes in 3D

The swept mesher operates on a 3D subdomain by meshing a source face and then sweeping the resulting face mesh along the subdomain to an opposite target face. For straight and circular sweep paths you can use several connected faces as source faces. Each face about a subdomain that is to be operated on by the swept mesher is classified as either a source face, a target face, or a boundary face. The boundary faces are the faces linking the source and target face.



The faces about a subdomain that is to be operated on by the swept mesher is classified as either a source face, a target face, or a boundary face.

The swept mesher can handle subdomains with multiple boundary faces in the sweep direction.

REQUIREMENTS ON THE GEOMETRY

For the sweeping technique to work, the geometry must satisfy these criteria:

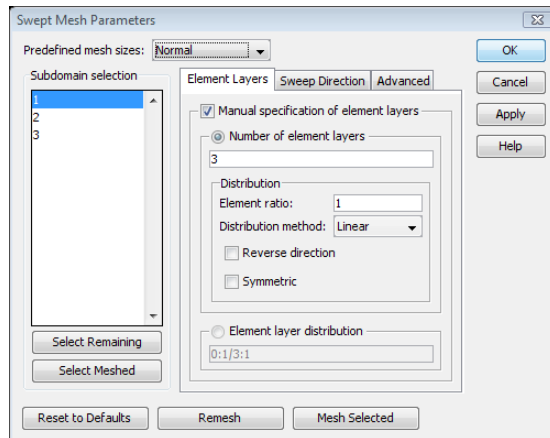
- Each subdomain must be bounded by one shell; that is, a subdomain must not contain holes except if they penetrate both the source and target face.
- There can only be one target face per subdomain. For straight and circular sweep paths several connected source faces are allowed. For other more general sweep paths there can only be one source face per subdomain.
- The source and target for a subdomain must be opposite each other in the subdomain's topology.
- The cross section along the direction of the sweep for a subdomain must be topologically constant.

USING THE SWEPT MESH PARAMETERS DIALOG BOX

To create a swept mesh on a 3D geometry use the **Swept Mesh Parameters** dialog box, which opens from the **Mesh** menu. Click the **Remesh** button to delete the current mesh and create a swept mesh on all subdomains as defined by the parameters in the dialog box. By clicking the **Reset to Defaults** button, you reset all parameter changes on all tabbed pages to their default values.

Following a click on the **Mesh Selected** button, the swept mesher creates a mesh only for the selected subdomains. You can read more on this in “Interactive Meshing” on page 346.

In the **Predefined mesh sizes** list you can select a setting that determines the size of the elements in the mesh. Predefined mesh sizes in the list include: extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, or extremely coarse.



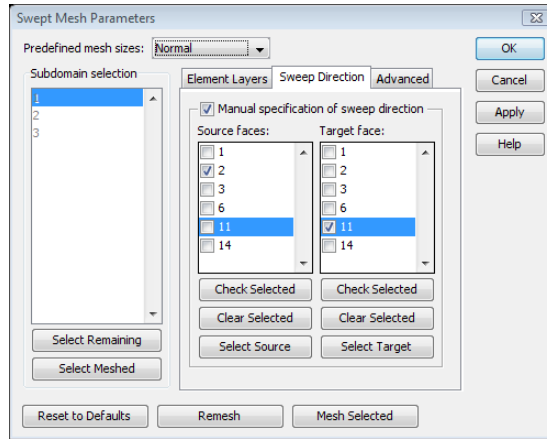
Selecting the Remaining Domains or the Meshed Domains

Click the **Select Remaining** button to select all subdomains that have not yet been meshed, or click **Select Meshed** to select all meshed subdomains.

Specifying Element Layers, Sweep Direction, and Advanced Parameters

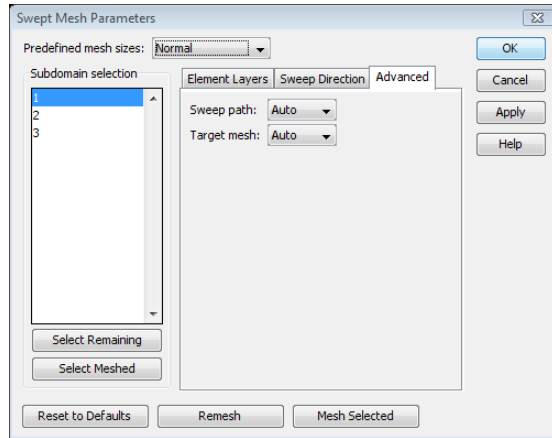
To specify the element distribution in the sweep direction for the selected subdomains select the **Manual specification of element layers** check box on the **Element Layers** page. You specify the element layers in the same way as when extruding a 2D mesh. For more information on this topic see “Specifying the Element Layers” on page 329. For subdomains where the **Manual specification of element layers** check box is not selected, the number of elements in the sweep direction is determined from the setting in the **Predefined mesh sizes** list.

To specify the direction of the sweep for a subdomain, first select the subdomain, go to the **Sweep Direction** page, and select the **Manual specification of sweep direction** check box. Note that this page is available only when you select exactly one subdomain. The face selection lists **Source faces** and **Target face** contain the faces only in the selected subdomain. Select the source face or faces in the **Source faces** list and the target face in the **Target face** list.



If only the source faces or only the target face is specified for a subdomain, COMSOL Multiphysics attempts to determine the opposite face. For a subdomain with no specified source and target face, the algorithm automatically attempts to determine a suitable pair of opposite faces. If the algorithm finds several opposite face pairs, the face with lowest geometric degree and face number becomes the source face.

On the **Advanced** page you can specify the shape of the sweep path in the **Sweep path** list and the method to be used for transferring the source mesh to the target face in the **Target mesh** list.



From the **Sweep path** list you select either **Auto**, **Straight**, **Circular**, or **General**:

- **Straight** means that all interior mesh points are located on straight lines between the corresponding source and target points.
- **Circular** means that all interior mesh points are located on circular arcs between the corresponding source and target points.
- **General** means that the positions of the interior mesh points are determined by a general interpolation procedure.
- **Auto**, which is default, means that COMSOL Multiphysics automatically tries to determine if the sweep path is straight or circular. Otherwise, the general procedure is used.

In the **Target mesh** list you select either **Auto**, **Morph**, or **Rigid**:

- **Morph** means that the target mesh is created from the source mesh by a morphing technique.
- **Rigid** means that the target mesh is created by a rigid transformation of the source mesh.
- **Auto**, which is default, means that COMSOL Multiphysics automatically tries to determine a suitable method for creating the target mesh.

Note: The **Sweep path** setting cannot be **General** when sweeping from several source faces or when the **Target mesh** setting is **Rigid**. Furthermore, when sweeping from several source faces, the **Target mesh** setting cannot be **Morph**.

MESH ELEMENTS IN THE SWEPT MESH

When creating a swept mesh, COMSOL Multiphysics automatically uses the free mesher to create a mesh on the source face. This means that the source face has a triangular mesh, and the swept mesh therefore consists of prism elements. However, it is also possible to create a swept mesh with hexahedral elements. You can find more information on this topic in “Creating Swept Meshes with Hexahedral Elements” on page 357.

The following image shows the result of a swept mesh on a geometry with one subdomain.



Swept mesh

SWEEPING ON MULTIPLE SUBDOMAINS

If you want to create a swept mesh on several consecutive subdomains, you do not need to specify the sweep direction for each subdomain. Instead you only specify the sweep direction for the subdomain where the sweep starts. The swept mesher then processes the subdomains consecutively according to the sweep direction of the starting subdomain. The mesher automatically determines the sweep direction for each consecutive subdomain such that the source face of the subdomain is the same as the

target face of the previously processed subdomain. For more information on this, see “Creating Swept Meshes Interactively” on page 353.

Creating Boundary Layer Meshes

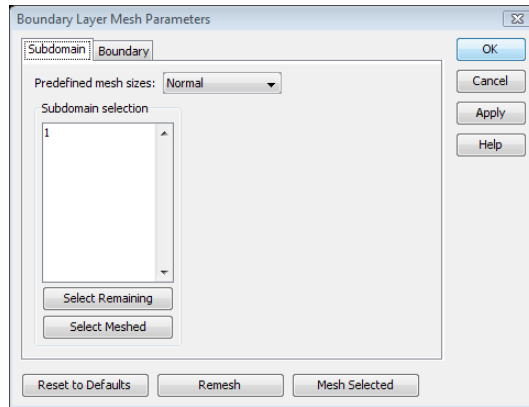
A boundary layer mesh is a mesh with dense element distribution in the normal direction along specific boundaries. This type of mesh is typically used for fluid flow problems to resolve the thin boundary layers along the no-slip boundaries. In 2D, a layered quadrilateral mesh is used along the specified no-slip boundaries. In 3D, the boundary layer mesh is a layered prism mesh or a hexahedral mesh depending on whether the corresponding boundary-layer boundaries contain a triangular or a quadrilateral mesh.

USING THE BOUNDARY LAYER MESH PARAMETERS DIALOG BOX

Use the **Boundary Layer Mesh Parameters** dialog box, which opens from the **Mesh** menu, to create a boundary layer mesh. Click the **Remesh** button to delete the current mesh and create a boundary layer mesh on all subdomains as defined by the parameters in the dialog box. By clicking the **Reset to Defaults** button, you reset all parameter changes on all tabbed pages to their default values.

When you create a boundary layer mesh using the **Remesh** button, the software automatically creates a free mesh according to the setting in the **Predefined mesh sizes** list. After that, it inserts boundary layer elements in the free mesh according to the boundary layer parameters specified on the **Boundary** page.

In the **Predefined mesh sizes** list you select a setting that determines the size of the elements in the initial free mesh. The available sizes are: extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, and extremely coarse.



To insert boundary layers to an existing mesh or to create a boundary layer mesh for a subset of all subdomains you select the corresponding subdomains and click the **Mesh Selected** button. You can read more on this topic in the section “Creating Boundary Layer Meshes Interactively” on page 359.

Selecting the Remaining Domains or the Meshed Domains

Click the **Select Remaining** button to select all subdomains that have not yet been meshed, or click **Select Meshed** to select all meshed subdomains.

Specifying the Boundary Layer Boundaries

Use the check boxes in the **Boundary selection** list on the **Boundary** page to choose the boundaries for which boundary layers will be created. The default value is to create boundary layers for all exterior boundaries.

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

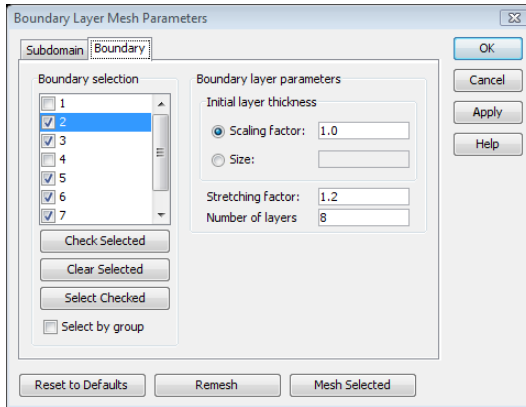
Specifying Boundary Layer Parameters

The right side of **Boundary** page contains an area where you can specify parameters for the boundary layers. Select one or multiple boundaries from the **Boundary selection** list and enter the desired parameters. It is possible to set parameters for any boundaries regardless of whether they are currently set to have boundary layer boundaries or not.

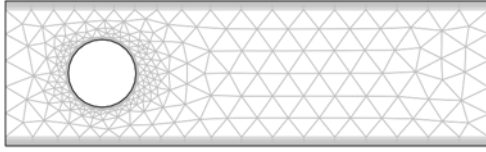
In the **Initial layer thickness** area you can specify the thickness of the first element layer, that is, the layer adjacent to the corresponding boundary. By default the thickness of

the first layer is $1/50$ of the size of the elements on the corresponding boundary. Use the **Scaling factor** edit field to specify a scaling factor that multiplies this default size. Alternatively, you specify the size of the first layer explicitly using the **Size** edit field.

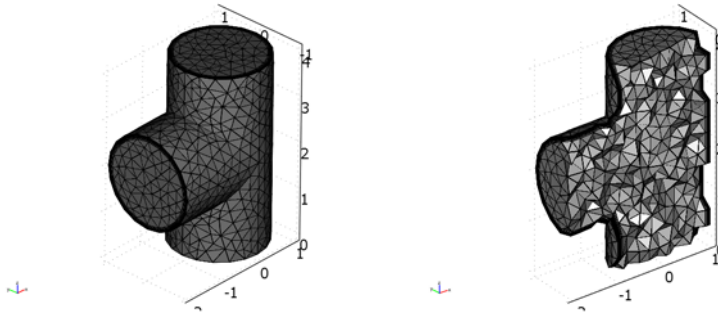
Use the **Stretching factor** edit field to specify the increase in thickness between two consecutive boundary layers, and use the **Number of layers** edit field to specify the total number of boundary layers.



The following images show examples of boundary layer meshes in 2D and 3D:

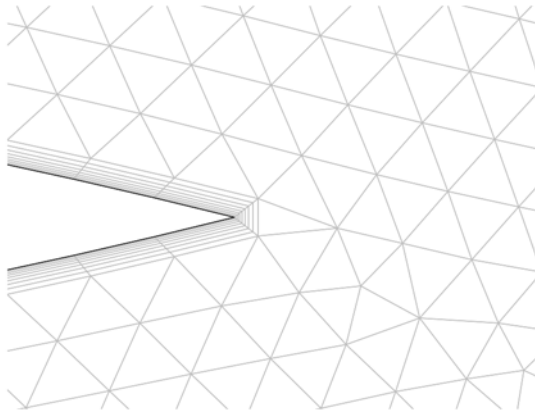


Boundary layer mesh in 2D based on a free mesh.



Boundary layer mesh in 3D based on a free mesh (full view to the left and cross-sectional view to the right).

In 2D, the mesher automatically introduces a boundary layer split at sharp corners to avoid boundary layer elements with bad quality. The next figure shows an example of a boundary layer split.



Boundary layer split at sharp corner.

Note: The mesher might adjust the number of boundary layers and the thickness of the boundary layers in thin regions to avoid low quality elements and colliding boundary layers.

MESH ELEMENTS IN THE BOUNDARY LAYER MESH

When creating a boundary layer mesh, COMSOL Multiphysics uses the free mesher to create an initial mesh in which the boundary layer elements are inserted. This means that the boundary layer mesh consists of triangular elements and quadrilateral elements in 2D and tetrahedral elements and prism elements in 3D. However, it is also possible to create a boundary layer mesh with only quadrilateral elements (in 2D) and a mesh with only hexahedral elements (in 3D). You can find more information on this topic in the section “Creating Boundary Layer Meshes Interactively” on page 359.

Meshing Thin Structures

Models with large geometric scale variations are sometimes problematic to mesh, in particular if they contain thin layers.

To begin with you should avoid using the free mesher on the thin domains. Instead use the mapped mesher in 2D and the swept mesher in 3D, if possible, because these meshers are less sensitive to thin regions. If you use the free mesher you can reduce the problems caused by the thin regions by meshing the thin domains separately and use the scale factors available in the **Scale geometry before meshing** area on the **Advanced** page in the **Free Mesh Parameters** dialog box. This reduces the scale differences during the meshing process and results in an anisotropic mesh on the thin domains. You can read more on how to mesh domains separately in “Interactive Meshing” on page 346.

In structural mechanics, specially devised *shell elements* for modeling thin structures have been around for many years. Typically, shell elements are hard-wired for structural mechanics or heat transfer problems. COMSOL Multiphysics offers the ability to model thin layers with equation-based modeling using tangential derivative variables. You can set up multicomponent PDEs on boundaries and edges using these variables, and even connect the boundary PDE model to PDEs on subdomains. More information about the use of tangential derivative variables appears in the chapter “The Weak Form” of the *COMSOL Multiphysics Modeling Guide*. The Structural Mechanics Module and the Heat Transfer Module contain specialized application modes for modeling of thin shells.

Mesh Statistics

The **Mesh Statistics** dialog box, available from the **Mesh** menu, contains statistical data for the current mesh such as the number of elements and the minimum element quality. On page 372 you can find an explanation of mesh quality measures. *Element volume ratio* (in 3D) and *element area ratio* (in 2D) are defined as the minimal element volume divided by the maximal element volume (in 3D) or the maximal element area (in 2D). If the model contains one or more application modes, the number of degrees of freedom for the model is available as well. For more information on the number of degrees of freedom see “Finite Elements” on page 482 in the *COMSOL Multiphysics Reference Guide*.

Use the domain pages named **Subdomain**, **Boundary**, **Edge**, and **Point** to get mesh statistics for individual domains. Use the mouse to select domains directly in the geometry or select domains from the domain selection lists on each page. You can also press Ctrl+A to select all subdomains, boundaries, edges, or points. The statistical data that you see applies to all selected domains of that type.

Refining Meshes

When trying to solve a problem that does not converge, or having solved a problem where the solution does not seem properly resolved, there is often a need for re-solving the problem using a finer mesh. In these situations it is sometimes easier to refine the existing mesh than to regenerate the mesh from scratch using finer mesh settings. It is only possible to refine free meshes containing triangular elements, or tetrahedral elements, in 2D and 3D, respectively.

Refinement Methods

To refine a free mesh go to the **Mesh** menu and choose **Refine Mesh**, or click the **Refine Mesh** button on the Main toolbar. In 1D, COMSOL Multiphysics always uses regular refinement, where it divides each element into two elements of the same shape. In 2D and 3D, however, two *refinement methods* are available: *regular* and *longest*. 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.

Select the refinement method in the **Free Mesh Parameters** dialog box by looking on the **Global** page for the **Refinement method** list. For 2D geometries COMSOL Multiphysics defaults to the regular refinement method, and in 3D the longest refinement method is the default.

Refining Selected Elements

In 2D, you can refine a subset of elements in a free triangle mesh using the mouse to select an area in which you want a finer mesh:

- 1 Click the **Refine Selection** button on the Main toolbar.
- 2 Using the mouse, drag to enclose the mesh elements that you want to refine.
- 3 COMSOL Multiphysics refines all mesh elements that the selection encloses completely and possibly also some additional elements to take care of hanging nodes at the boundary of the refined area.

Converting Mesh Elements

Some geometries have subdomains that are well suited for swept meshes. If there are surrounding subdomains that cannot be swept, it is necessary to convert the hexahedral or prism mesh in the swept subdomains to a tetrahedral mesh before using the free mesher on the surrounding subdomains. This is because it is impossible to create tetrahedrons adjacent to elements with quadrilateral faces.

Another reason to convert a mesh is that 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, prism, and hexahedral elements, to a pure tetrahedral mesh. The mesh conversion function 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.

Converting Elements

To convert every tetrahedral, prism, or quadrilateral element in the mesh to simplex elements, go to the **Mesh** menu and choose **Convert All** from the **Interactive Meshing** submenu, or click the **Convert All** button on the Mesh toolbar.

It is also possible to convert elements in a particular subdomain or, in 3D, face. Select the subdomains you want to convert and choose **Convert Selected** from the **Interactive Meshing** submenu or click the **Convert Selected** button on the Mesh toolbar. To convert quadrilateral elements on a particular face in 3D, select the faces you want to convert, and choose **Convert Selected** as above.

For information on the methods used to convert elements, see the documentation for the `meshconvert` function on page 291 of the *COMSOL Multiphysics Reference Guide*.

Interactive Meshing

The interactive meshing environment makes it possible to build a mesh in an incremental fashion where each meshing operation acts on a set of geometry domains. For example, you can start by creating a boundary mesh and then mesh each subdomain sequentially. This makes it easier to control the distribution of elements and the size of the mesh. Furthermore, using interactive meshing you can apply different meshing techniques to different domains of a geometry object.

Using the Mesh Toolbar

The Mesh toolbar is available in the selection modes and in Mesh mode. This toolbar contains buttons for all available meshing operations. All meshing tools that are available through the buttons are also available from the **Interactive Meshing** submenu on the **Mesh** menu.

Selecting the **Interactive Meshing** button on the top of the Mesh toolbar provides a display of the mesh in the selection modes such that you can see the domains that are meshed. Deselect the **Interactive Meshing** button to turn off the visualization of the mesh in the selection modes.

















The Mesh toolbar consists of four groups of buttons:








- The first group corresponds to meshing tools that operate on the entire geometry or the part of the geometry that is currently not meshed; these operations are not affected by the current domain selection.
- The second group of buttons corresponds to meshing tools that operate only on the selected domains.
- The third group contains two buttons: one for selecting the remaining domains, that is, the domains that have not yet been meshed, and one for selecting the domains that have already been meshed.
- The fourth group, with two buttons, controls the predefined settings that determine the size of the mesh when creating one.

Clicking a button from the second group in the Mesh toolbar, that creates a mesh, is analogous to clicking the **Mesh Selected** button in the corresponding dialog box.

THE MESH TOOLBAR IN 3D





The Mesh toolbar in 3D contains the following buttons:

-  • **Interactive Meshing**—Select this toggle button to enable visualization of the mesh in the selection modes.
-  • **Mesh All (Free)**—Click this button to delete the current mesh and create a free mesh on the entire geometry.
-  • **Mesh All (Swept)**—Click this button to delete the current mesh and create a swept mesh on all subdomains.
-  • **Mesh All (Boundary Layer)**—Click this button to delete the current mesh and create a boundary layer mesh on all subdomains.
-  • **Mesh Remaining (Free)**—Click this button to create a free mesh on the part of the geometry that is not meshed.
-  • **Mesh Remaining (Swept)**—Click this button to create a swept mesh on the subdomains that are not meshed.
-  • **Mesh Remaining (Boundary Layer)**—Click this button to create a boundary layer mesh on the subdomains that are not meshed.
-  • **Convert All**—Click this button to convert the entire mesh to a simplex mesh.
-  • **Refine Mesh**—Click this button to refine the entire mesh.
-  • **Delete Mesh**—Click this button to delete the entire mesh.
-  • **Mesh Selected (Free)**—Click this button to create a free mesh on the selected domains.
-  • **Mesh Selected (Free, Quad)**—Click this button to create a free quad mesh on the selected boundaries. This button is available only in Boundary mode.
-  • **Mesh Selected (Mapped)**—Click this button to create a mapped quad mesh on the selected boundaries. This button is available only in Boundary mode.
-  • **Mesh Selected (Swept)**—Click this button to create a swept mesh on the selected subdomains. This button is available only in Subdomain mode.
-  • **Mesh Selected (Boundary Layer)**—Click this button to create a boundary layer mesh on the selected subdomains. This button is available only in Subdomain mode.
-  • **Convert Selected**—Click this button to convert the mesh on the selected subdomains or faces to a simplex mesh.

- 
• **Refine Selected**—Click this button to refine the mesh on the selected subdomains. This button is available only in Subdomain mode.
- 
• **Copy Mesh**—Click this button to copy the boundary mesh from the selected meshed boundaries to the selected unmeshed boundary. This button is available only in Boundary mode.
- 
• **Delete Selected**—Click this button to delete the mesh on the selected domains.
- 
• **Select Remaining**—Click this button to select the domains on the current domain level that are not meshed.
- 
• **Select Meshed**—Click this button to select the meshed domains on the current domain level.
- 
• **Increase Mesh Size**—Click this button to increase the **Predefined mesh sizes** setting, for example, from normal to coarse.
- 
• **Decrease Mesh Size**—Click this button to decrease the **Predefined mesh sizes** setting, for example, from normal to fine. The tooltips show the mesh sizes for the mesh before and after increasing or decreasing the mesh size and the status bar shows the current mesh size.

THE MESH TOOLBAR IN 2D

The Mesh toolbar in 2D contains almost the same tools as in 3D. However, in 2D the the following buttons for creating mapped and free quad meshes replace the buttons for creating swept meshes in 3D:

- 
• **Mesh All (Mapped)**—Click this button to delete the current mesh and create a mapped quad mesh on all subdomains.
- 
• **Mesh Remaining (Mapped)**—Click this button to create a mapped quad mesh on the subdomains that are not meshed.
- 
• **Mesh Selected (Mapped)**—Click this button to create a mapped quad mesh on the selected subdomains. This button is available only in Subdomain mode.
- 
• **Mesh All (Free, Quad)**—Click this button to delete the current mesh and create a free quad mesh on the entire geometry.



• **Mesh Remaining (Free, Quad)**—Click this button to create a free quad mesh on the subdomains that are not meshed.



• **Mesh Selected (Free, Quad)**—Click this button to create a free quad mesh on the selected subdomains. This button is available only in subdomain mode.

The Mesh toolbar in 2D contains the following additional buttons:



• **Extrude Mesh**—Click this button to extrude the entire mesh into 3D.



• **Revolve Mesh**—Click this button to revolve the entire mesh into 3D.



• **Refine Selection**—Click this button to refine a selection of elements in the mesh. This button is available only in the Subdomain selection mode.

THE MESH TOOLBAR IN 1D

Because the mapped mesher, free quad mesher, swept mesher, and boundary layer mesher are not available in 1D, the Mesh toolbar in 1D contains a subset of the tools available in the Mesh toolbar for 2D and 3D.

Deleting the Mesh

To delete the mesh on the entire geometry click the **Delete Mesh** button on the Mesh toolbar. To delete the mesh on a specific domain, select the domain and click the **Delete Selected** button. This also deletes the mesh on the domain's boundary.

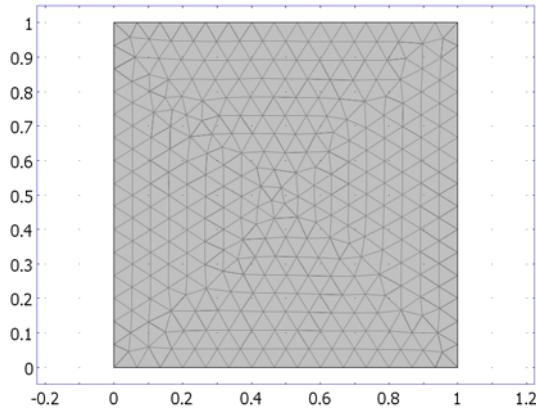
Meshing a Domain Without Remeshing Boundaries

It is possible to mesh a domain that is already meshed without first deleting its mesh. When you do this, the boundary mesh is not affected. As the next example shows, you can use this method to mesh a domain with coarse mesh settings but keep the fine boundary mesh.

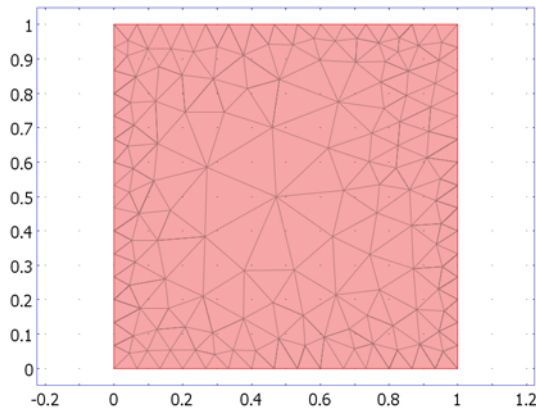
If you want to remesh the boundary of the domain as well, you must delete the mesh on the domain before meshing it again.

EXAMPLE OF MESHING A DOMAIN AGAIN

- 1 Create a free mesh (with the default settings) on a unit square geometry using the **Mesh All (Free)** tool on the Mesh toolbar.

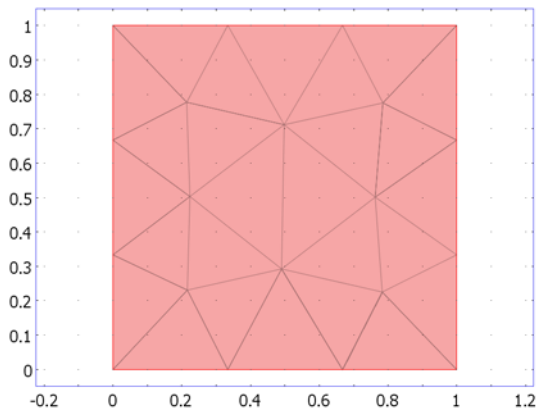


- 2 Change the setting for the predefined mesh size from Normal to Extremely coarse by clicking the **Increase Mesh Size** button on the Mesh toolbar four times.
- 3 Select the subdomain and click the **Mesh Selected (Free)** button on the Mesh toolbar. Notice that the boundary is not remeshed.



- 4 Delete the mesh on the subdomain by selecting the subdomain and clicking **Delete Selected**.

- 5 Create a new free mesh by selecting the subdomain and clicking **Mesh Selected (Free)**.
Now the entire geometry is meshed using the extremely coarse mesh setting.



If you want to remesh a domain that is adjacent to any other meshed domain of higher dimension you must also delete the mesh on the domain before meshing it again.

Undoing Meshing Operations

It is possible to undo all operations available from the Mesh toolbar except for the **Extrude Mesh** and **Revolve Mesh** operations. To undo a meshing operation, choose **Undo Mesh** from the **Edit** menu or press Ctrl+Z.

There is a limit to the number of mesh operations you can undo. The default is three. To change this value, open the **Preferences** dialog box from the **Options** menu, find the **Maximum number of mesh undo steps** edit field, then enter the desired number of undo steps. Note that COMSOL Multiphysics stores the previous meshes in memory, so we recommend that you use a small number of undo steps.

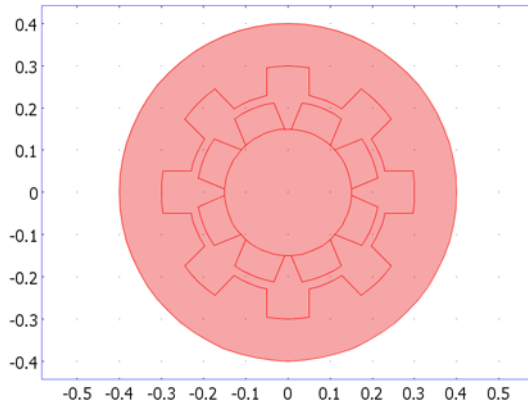
Creating a Combined Free and Mapped Mesh in 2D

Using the interactive meshing tools it is possible to combine a free triangular, free quadrilateral, and mapped quadrilateral mesh in the same geometry. If you want to create such a combined mesh, it is often a good choice to start meshing the domains where the mapped quad mesh is desired. This is because the mapped quad mesher requires matching edge meshes on opposite sides of the subdomains. For more information on the mapped quad mesher see “Creating Mapped Meshes in 2D” on page 320. Each subdomain that is to be operated on by the free quad mesher needs to

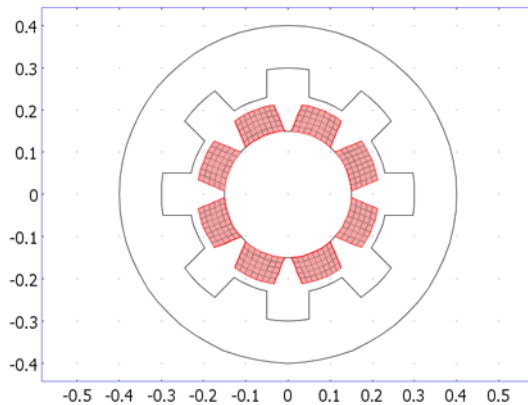
be surrounded by an even number of edge elements. It is therefore a good idea to mesh these domains second, leaving the domains to be meshed with the free triangle mesher to last.

EXAMPLE OF A COMBINED FREE AND MAPPED MESH

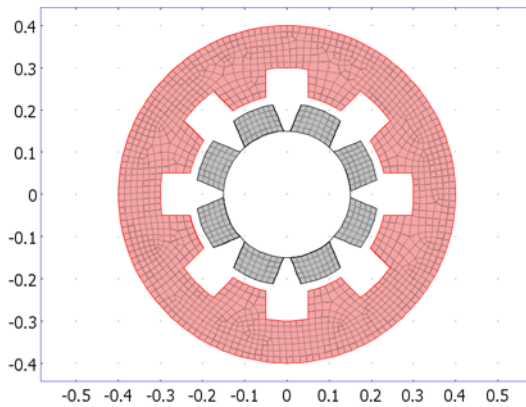
The geometry in the figure below consists of eleven subdomains. The following example shows how to mesh some of these subdomains with a mapped quad mesh, some with a free quad mesh, and the rest with a free triangle mesh.



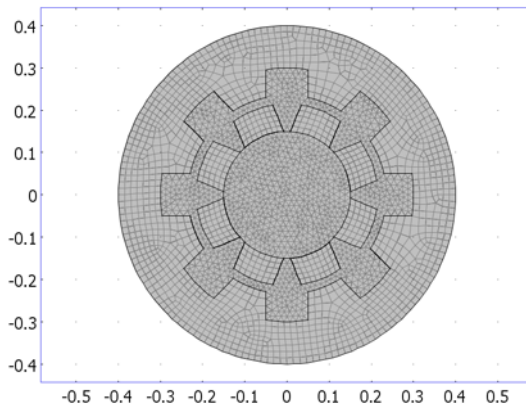
- I Start to create a mapped mesh on eight of the subdomains by selecting the subdomains, and clicking the **Mesh Selected (Mapped)** button.



- 2 Mesh the outermost subdomain with an unstructured quad mesh by selecting the subdomain, and clicking the **Mesh Selected (Free, Quad)** button.



- 3 Mesh the remaining subdomains with an unstructured triangle mesh by clicking the **Mesh Remaining (Free)** button.



Creating Swept Meshes Interactively

Instead of specifying the element-layer distribution and the sweep direction from the **Swept Mesh Parameters** dialog box, you can mesh one of the edges in the sweep direction and the source face before creating the swept mesh. For more information on the swept mesher see “Creating Swept Meshes in 3D” on page 333.

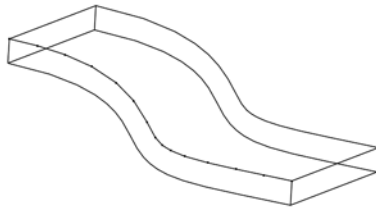
MESHING AN EDGE IN THE SWEEP DIRECTION BEFORE SWEEPING

By meshing one or more edges in the sweep direction before creating the swept mesh, the mesh on the edge(s) is used when determining the element-layer distribution for the swept mesh. This is particularly useful if you want to create a swept mesh for a subdomain where the sweep path is curved.

As an example, create a swept mesh on the geometry in the following figure:

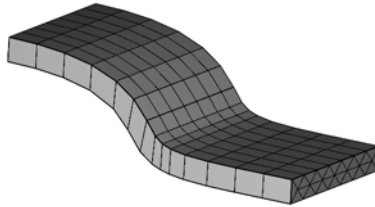


- 1 Before creating the swept mesh, mesh an edge along the sweep path to get a guiding mesh for the swept mesher.



- 2 Create a swept mesh by selecting the subdomain and clicking **Mesh Selected (Swept)**. The swept mesher uses the predefined edge mesh when determining the

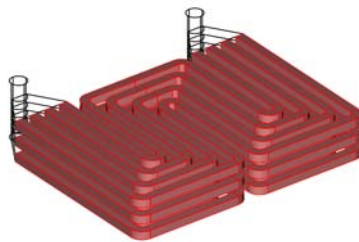
element-layer distribution in the sweep direction. This way you get a swept mesh that is adjusted to the curvature of the sweep path.



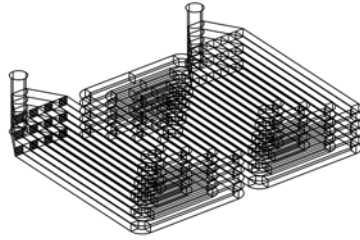
MESHING THE SOURCE FACE BEFORE SWEEPING

Instead of specifying the sweep direction for a subdomain in terms of a source face, you can simply mesh the corresponding face before creating the swept mesh. The swept mesher then uses this face as the source face as long as you do not explicitly specify any other sweep direction. This is particularly useful when you want to create a swept mesh on several groups of consecutive subdomains.

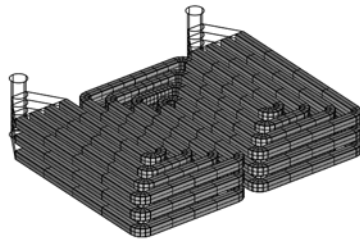
As an example, create a swept mesh on the selected subdomains in the geometry shown in the following image. These subdomains define 12 independent sweep paths with several subdomains in each path.



- 1 Mesh the source faces for the sweep using the free mesher by selecting the corresponding faces and clicking the **Mesh Selected (Free)** button on the Mesh toolbar.



- 2 Select all subdomains to be swept and click the **Mesh Selected (Swept)** button to create the swept mesh. The faces meshed prior to the mesh-sweeping operation serve as source faces for the sweep. Furthermore, the sweep direction for each consecutive subdomain is automatically determined such that the sweep follows the direction outlined by the corresponding source face.



CREATING COMBINED FREE AND SWEEP MESHES IN 3D

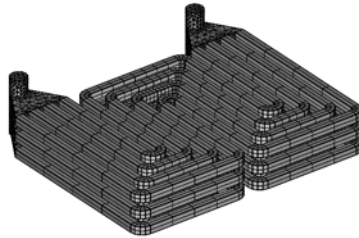
You can combine a free mesh and a swept mesh in the same geometry as long as the faces between subdomains with different element types are meshed consistently. For example, all faces in a subdomain to be meshed with the free mesher must contain only

triangular elements, and the boundary faces of a subdomain to be meshed with the swept mesher must contain mapped quad meshes.

Example, Continued

Continue meshing the previous example.

- 3 Create a free mesh on the remaining domains by clicking the **Mesh Remaining (Free)** button.



CREATING SWEPT MESHES WITH HEXAHEDRAL ELEMENTS

When creating a swept mesh on a subdomain without meshing the source face before using the swept mesher, COMSOL Multiphysics automatically uses the free mesher to mesh the source face. This results in a swept mesh consisting of prism elements. However, by meshing the source face with either the free or the mapped quad mesher before creating the swept mesh, you get a swept mesh consisting of hexahedral elements.

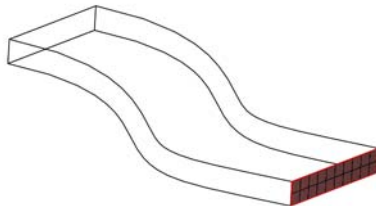
To create a mapped quad mesh on a face, select the face and click the **Mesh Selected (Mapped)** button. Alternatively, use the **Mapped Mesh Parameters** dialog box that works in the same way as in 2D (for more information see “Creating Mapped Meshes in 2D” on page 320). To create a free quad mesh on a face, select the face and click the **Mesh Selected (Free, Quad)** button, or use the **Free Mesh Parameters** dialog box. On the **Boundary** tab select the faces to be meshed, choose **Quad** from the **Method** list, and click **Mesh Selected**. In 3D the free quad mesher and the mapped quad mesher are available only in Boundary mode.

Example—Swept Mesh With Hexahedral Elements

Create a swept mesh with hexahedral elements on the geometry shown in the following figure.



- 1 Mesh the source face of the sweep using the mapped quad mesher by selecting the corresponding face and clicking the **Mesh Selected (Mapped)** button.



- 2 Create a swept mesh by selecting the subdomain and clicking **Mesh Selected (Swept)**.

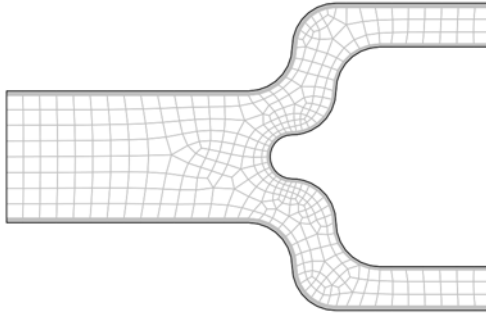


Creating Boundary Layer Meshes Interactively

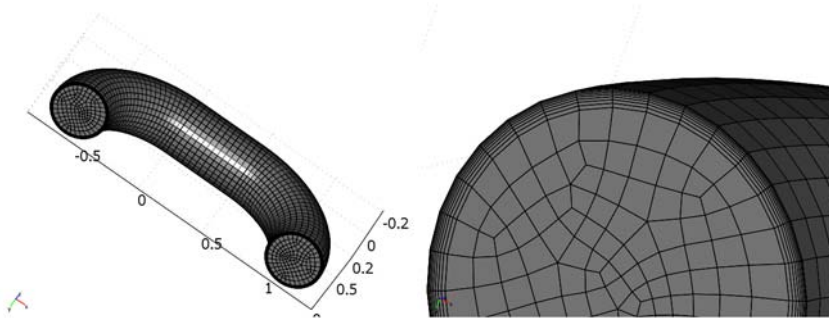
If you click the **Remesh** button in the **Boundary Layer Mesh Parameters** dialog box or the **Mesh All (Boundary Layer)** button on the Mesh toolbar, COMSOL Multiphysics deletes the entire mesh and uses the free mesher to create a new initial mesh in which the program then inserts boundary layers. For more information on the boundary layer mesher, see the section “Creating Boundary Layer Meshes” on page 338.

It is also possible to insert boundary layers into an existing mesh. You do this by selecting the subdomains to be processed and clicking the **Mesh Selected** button in the **Boundary Layer Mesh Parameters** dialog box or by clicking the **Mesh Selected (Boundary Layer)** button on the Mesh toolbar.

The following images show two meshes where boundary layers have been inserted in a free quad mesh in 2D and in a swept hex mesh in 3D.



Boundary layers inserted in a free quad mesh.



Boundary layers inserted in a swept hexahedral mesh (full view to the left and zoomed view to the right).

Copying Meshes

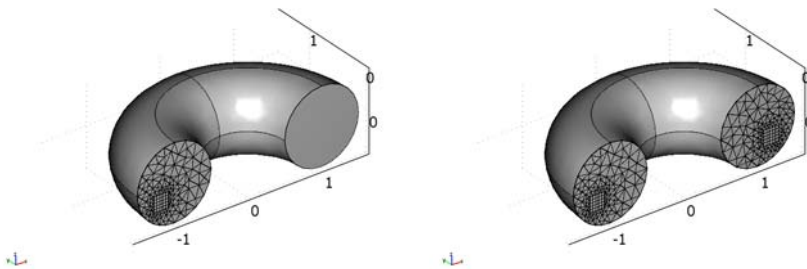
In most situations where it is necessary to couple the solution variables between boundaries, using identical meshes on the boundaries can result in significant gains. This is the case, for instance, when coupling across assembly interfaces or for models with periodic boundaries. The identical mesh improves accuracy and eliminates the need for extra degrees of freedoms in some problems, for example, in electromagnetic models using vector elements. COMSOL Multiphysics provides an easy tool for creating identical boundary meshes through its mesh copying functionality. It is possible to copy meshes in both 2D and 3D.

USING THE COPY MESH TOOLBAR BUTTON

The easiest way to copy a mesh between boundaries is to use the **Copy Mesh** button on the Mesh toolbar. Select the source boundary or boundaries from which to copy the mesh and the single target boundary where make a copy, and then click the **Copy Mesh** button. The software automatically determines how to orient the source mesh on the target boundary when you copy the mesh using this button. Note that all the source boundaries must be meshed and the target boundary unmeshed when you click the **Copy Mesh** button. Also, copying is only possible between boundaries that have the same shape. However, a scaling factor between the boundaries is permissible.

The copy mesh algorithm automatically tries to determine the best mapping from the source boundaries to the target boundary based on geometric properties. However, for complicated geometries the algorithm might not be able to determine a mapping between the boundaries. In such cases, you need to specify the mapping explicitly using the **Copy Boundary Mesh Parameters** dialog box.

The following images show a geometry where the mesh has been copied from the two source boundaries on the left side to the single boundary on the right side using the **Copy Mesh** toolbar button. Note that copying in the opposite direction would not be possible, because copying from one to several boundaries is not supported.



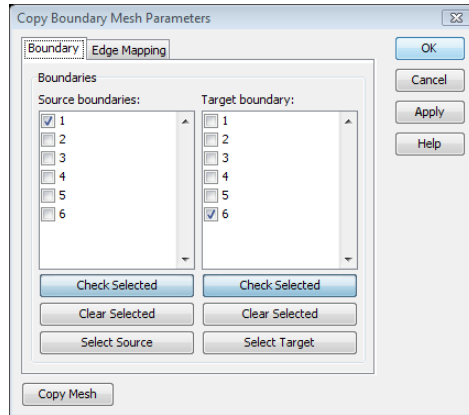
Boundary mesh copied from the two boundaries on the left to the single boundary on the right.

USING THE COPY BOUNDARY MESH PARAMETERS DIALOG BOX

The **Copy Boundary Mesh Parameters** dialog box, which opens from the **Mesh** menu, is available in 2D and 3D.

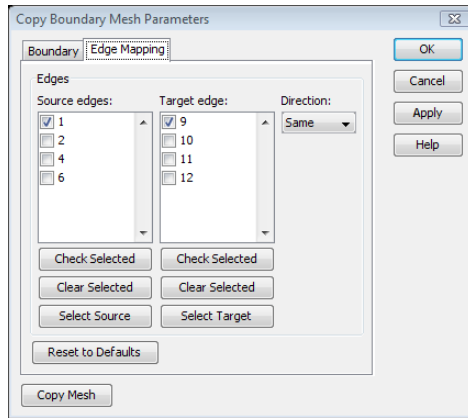
In 3D, the dialog box contains two pages: **Boundary** and **Edge Mapping**. On the **Boundary** page, you select the source boundary or boundaries from where to copy the mesh and a single target boundary. When you then click the **Copy Mesh** button, the

source mesh is copied to the target boundary using the default setting for the mapping between the boundaries. Note that all the source boundaries must be meshed before the copy operation. The target boundary can be either unmeshed or meshed; in the latter case, the existing mesh is first deleted, and the source mesh is then copied to the target boundary.

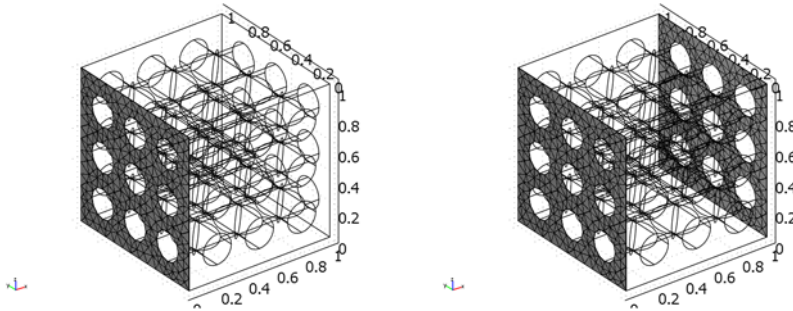


If the mesh generated by the default setting is not what you wanted, or if the algorithm was unable to determine the relation between the boundaries, go to the **Edge Mapping** page to explicitly specify the mapping. When the page opens, the default mapping is automatically selected. If the algorithm fails to determine a default mapping, all the check boxes are cleared. Select one or several source edges, a single target edge, and the direction between the edges. If there are several source edges, the direction refers to the direction between the target edge and the source edge with the lowest edge index. Choose **Auto** to let the algorithm automatically determine a suitable direction

between the edges. Click **Copy Mesh** on the **Boundary** page to copy the boundary mesh using the specified settings.



The following images show a periodic geometry where the mesh on the left-side boundary is copied to the right-side boundary. Note that creating a swept mesh is not possible for this geometry, so the copy mesh functionality is the only way to create an identical mesh on both boundaries.



Boundary mesh copied between periodic boundaries.

In 2D, the **Copy Boundary Mesh Parameters** dialog box contains only one page on which you specify the source boundaries from where to copy the mesh, the target boundary to copy the mesh to, and the direction between the boundaries. The direction refers to the direction between the target boundary and the source boundary with the lowest edge index.

Importing and Exporting Meshes

Importing a mesh into COMSOL Multiphysics completely eliminates the need for drawing, importing, or otherwise manipulating a geometry. This makes it possible to directly use a mesh that was created with an external program. COMSOL Multiphysics supports the most common 3D mesh format: the *NASTRAN bulk data file*. You can also import other mesh formats by translating them into this mesh format or into a *COMSOL Multiphysics file*.

Using COMSOL Multiphysics Files

It is possible to import or export a mesh object using a COMSOL Multiphysics file, which can exist in binary format (extension `.mphbin`) and text format (extension `.mphtxt`). More information on these file formats is available in the chapter “The COMSOL Multiphysics Files” in the *COMSOL Multiphysics Reference Guide*.

You export a mesh object to a COMSOL Multiphysics file by choosing **Export>Mesh To File** from the **File** menu. To import a mesh stored in this file format, choose **Import>Mesh From File** from the **File** menu.

USING A TEXT FILE TO IMPORT EXTERNAL MESH DATA

You can also use a COMSOL Multiphysics text file to transfer externally generated mesh data into a mesh object. The easiest way to do so is to first export a mesh object to a COMSOL Multiphysics text file; then replace the data in the different entries in that file with the corresponding external data; finally import the resulting file. An alternative approach of importing external mesh data is described in the chapter “Creating Meshes” in the *COMSOL Multiphysics Scripting Guide*.

Importing NASTRAN Files

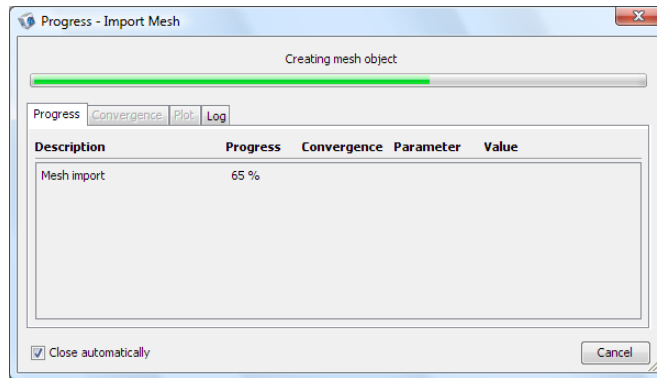
You can import 3D meshes in the NASTRAN bulk data format, the most common format for exchanging 3D meshes among programs. This format supports hundreds of NASTRAN entries describing elements, loads, and materials, making it possible to define a complete finite element model. When you import a NASTRAN bulk data file into COMSOL Multiphysics, the software imports only true mesh element information. You can let COMSOL Multiphysics determine the partitioning of the imported mesh elements into different domains according to material information in the file, if material information is available. The `meshimport` reference entry on page

307 of the *COMSOL Multiphysics Reference Guide* contains a list of the entries processed by the software.

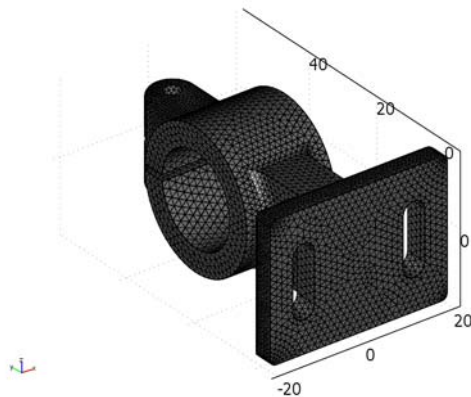
EXAMPLE OF IMPORT FROM NASTRAN BULK DATA FORMAT

To import the file `nastrandemo1.nas`, follow these steps:

- 1 Open the **Import Mesh** dialog box by going to the **File** menu and choosing **Import, Mesh From File**.
- 2 In the **Files of type** list select **NASTRAN File**.
- 3 Locate the Comsol installation directory, go to the **demo** folder, then select the file `nastrandemo1.nas`.
- 4 Click **Import** to close the dialog box and import the mesh in the file. This action also opens the **Import Mesh** progress window, which displays the current progress.



The next figure shows the imported mesh.



ELEMENT ORDER AND SHAPE FUNCTIONS

Mesh elements stored in the NASTRAN bulk data format are either linear or of the second order. However, you can select the order of the shape functions for the elements in the COMSOL Multiphysics model independently of the order of the elements in the imported mesh object. If you have imported a mesh with second-order mesh elements, the interpolation to the higher-order node points of the COMSOL Multiphysics elements uses the information about the second-order node points in the imported mesh. For details about the interpolation, see `meshimport` reference entry on page 307 of the *COMSOL Multiphysics Reference Guide*.

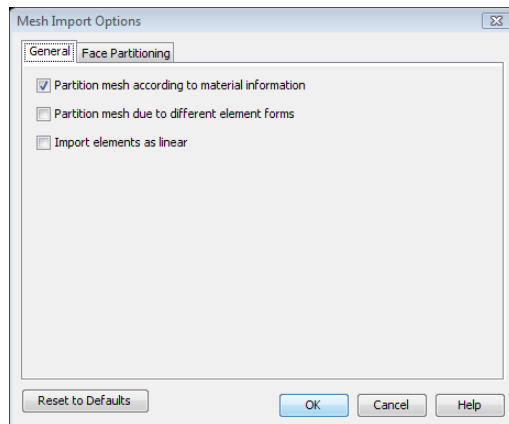
It is possible to import all elements as linear mesh elements regardless of their order in the NASTRAN bulk data file by going to the **Mesh Import Options** dialog box and selecting the **Import elements as linear** check box on the **General** page.

CREATING DOMAIN INFORMATION

Generally, a NASTRAN bulk data file contains only solid elements. This means that COMSOL Multiphysics enriches the imported mesh data with boundary elements, edge elements, and vertex elements such that a valid mesh object is formed. Each element in the imported mesh object also gets a unique geometry domain label. You can control the grouping of the imported elements into different domains by opening the **Mesh Import Options** dialog box from the **Import Mesh** dialog box.

Subdomain Level

On the **General** page you control the domain partitioning at the subdomain level.



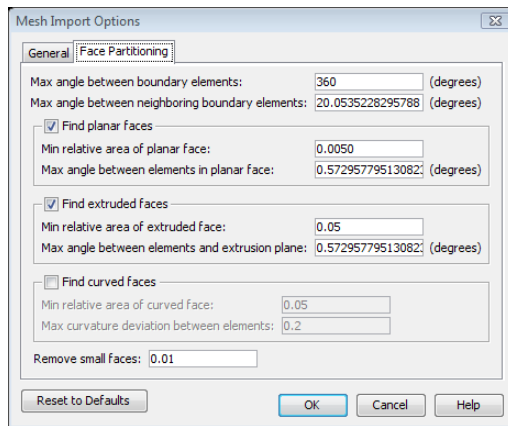
- The **Partition mesh according to material data** check box controls if material data in the NASTRAN bulk data file is used (if available) to determine the domain

partitioning of the subdomain elements. If you clear this check box, all subdomain elements in the imported mesh object belong to the same subdomain, if possible.

- With the **Partition mesh due to different element forms** check box you specify if mesh elements of different element forms—that is, tetrahedral, pentahedral, or hexahedral—get different subdomain labels.

Boundary Level

On the **Face Partitioning** page you have access to several properties that influence the domain partitioning at the boundary level. Note that the domain partitioning of the subdomain elements induces an initial domain partitioning of the boundary elements.



- In the **Max angle between boundary elements** edit field you specify the maximum allowed angle between any two boundary elements in the same face.
- In the **Max angle between neighboring boundary elements** edit field you specify the maximum angle between a boundary element and a neighbor that causes the boundary elements to be part of the same face, if possible.
- In the **Find planar faces** area you specify parameters for detecting planar faces. Inactivate the search for planar faces by clearing the check box.
 - In the **Min relative area of planar face** edit field you specify the minimum relative area that a planar face can have to be created as a new face.
 - In the edit field **Max angle between elements in planar face** you specify the maximum allowed angle between two boundary elements to be part of the same planar face.
- In the **Find extruded faces** area you specify parameters for detecting extruded faces. An extruded face is defined by a set of boundary elements where all element normals

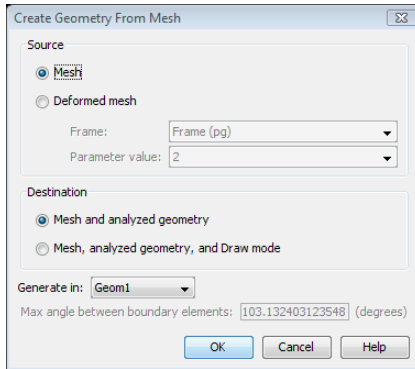
lie in the same plane, referred to as *extrusion plane*. Inactivate the search for extruded faces by clearing the check box.

- In the **Min relative area of extruded face** edit field you specify the minimum relative area that an extruded face can have to be created as a new face.
- In the **Max angle between elements and extrusion plane** edit field you specify the maximum angle between a boundary element normal and the extrusion plane that will cause the element to be part of the extruded face if possible.
- In the **Find curved faces** area you specify parameters for detecting faces with a constant curvature. This property is active only for meshes with 2nd-order elements. Inactivate the search for faces with constant curvature by clearing the check box.
 - In the **Min relative area of curved face** edit field you specify the minimum relative area that a curved face is allowed to have to be created as a new face.
 - In the **Max curvature deviation between elements** edit field you specify the maximum relative curvature difference between boundary elements that will cause the elements to be part of the same curved face if possible.
- In the **Remove small faces** edit field you specify a positive scalar controlling to which extent the software merges small faces with adjacent faces. If you get an imported mesh object with exceedingly small faces that you want to get rid of, increase the value of this property and re-import the file.

Note: When importing a mesh to COMSOL Multiphysics, there is no corresponding geometry object available. This means that it is not possible to refine an imported mesh, and you cannot use the adaptive solver unless you create a geometry object from the imported mesh.

Creating a Geometry from a Mesh

A geometry is necessary for remeshing and when using the adaptive solver, but an imported mesh has no corresponding analyzed geometry. You can generate one from a mesh by choosing **Create Geometry from Mesh** from the **Mesh** menu.



In the **Source** area, select whether to create a geometry from a **Mesh** or from a **Deformed mesh** generated by solving a parameterized geometry or ALE problem. To create a geometry from a deformed mesh, you must also specify the frame and solution to use.

In the **Destination** area, select the **Mesh and analyzed geometry** option to create an analyzed geometry, or select the **Mesh, analyzed geometry, and Draw mode** option to create an analyzed geometry and a Draw mode object. Both options include mesh as destination because the software may introduce new boundaries during the creation and then needs to update the mesh to be compatible with the new geometry.

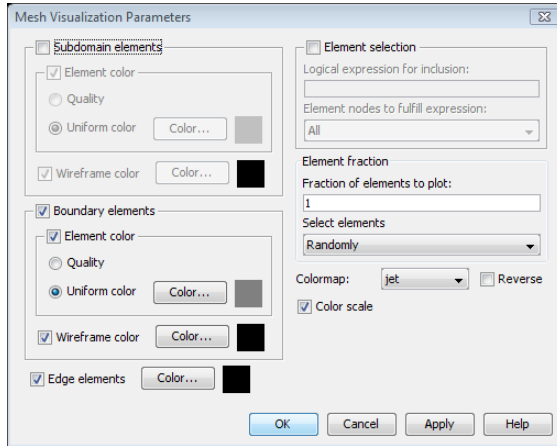
You can dictate on which geometry the generated analyzed geometry is created with the **Generate in** list. The default destination is the current geometry, but you can choose any other geometry in the model or create a new geometry with the generated analyzed geometry.

In the **Max angle between boundary elements** edit field you specify the maximum allowed angle between two boundary elements to be part of the same face. This parameter is available only if the mesh has no parameterization.

When you generate an analyzed geometry from a 3D mesh and the **Max angle between elements in planar face** parameter is available, the generation may introduce new boundaries.

Visualizing Meshes

You can visualize a mesh in several ways. Mesh visualizations are carried out in Mesh mode, and related options are available in the **Mesh Visualization Parameters** dialog box.



Using the **Subdomain elements** check box, specify whether or not subdomain elements should be visible in the mesh plot. In the **Subdomain elements** area, specify what should be visible: the interior of the elements (using the **Element color** check box), just the wireframe (using the **Wireframe color** check box), or both. For each option you can set the color using the corresponding **Color** button. In addition, you can color the interior of the subdomain elements according to the element quality using the **Quality** option.

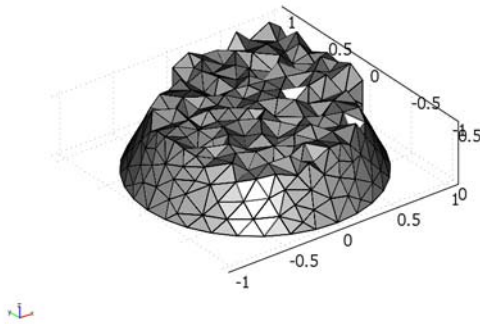
Using the **Boundary elements** check box, specify whether or not boundary elements should be visible in the mesh plot. Use the corresponding **Color** button to set their color. Alternatively, you can color the boundary elements according to the element quality using the **Quality** option.

In 3D, the **Edge elements** check box specifies whether or not edge elements should be visible in the mesh plot. Use the corresponding **Color** button to set their color.

In the **Element selection** area you control which part of the mesh to make visible. This is most useful in 3D, where this option gives a method of “seeing inside” the mesh.

In the **Element fraction** area you specify what fraction of subdomain elements to show. Enter a number between 0 and 1 in the **Fraction of elements to plot** edit field, then select random elements or those with best or worst quality in the **Select elements** list.

The figure below shows a mesh plot with only subdomain elements at $z < 0.5$ visible.

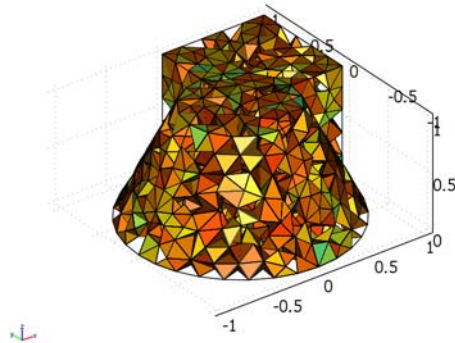


Mesh plot with only the elements at $z < 0.5$ visible

Visualizing the Mesh Quality

To visualize the mesh element quality, use the **Quality** setting in the **Mesh Visualization Parameters** dialog box. In 2D it is also possible to get a plot of the mesh quality for all subdomain elements by choosing **Display Element Quality** from the **Mesh** menu. The variable `qual` is available in postprocessing for inclusion of the mesh quality in plot expressions, for example, in the **Plot Parameters** dialog box.

The following figure shows a plot of mesh quality where only a fraction of the subdomain elements, with the worst quality, is visible.



Element quality plot for a fraction of the elements

MESH QUALITY MEASURES

The quality measure is related to the aspect ratio, which means that anisotropic elements can get a low quality measure even though the element shape is reasonable. For triangular elements, COMSOL Multiphysics computes the mesh quality q as

$$q = \frac{4\sqrt{3}A}{h_1^2 + h_2^2 + h_3^2},$$

where A is the area, and h_1 , h_2 , and h_3 are the side lengths of the triangle.

The value of q is a number between 0 and 1. If $q > 0.3$, the mesh quality should not affect the solution's quality; $q = 1$ when $h_1 = h_2 = h_3$. For a quadrilateral,

$$q = \frac{4A}{h_1^2 + h_2^2 + h_3^2 + h_4^2},$$

where h_1 , h_2 , h_3 , and h_4 are the side lengths. For a square, $q = 1$.

For tetrahedral elements, COMSOL Multiphysics computes the mesh quality as

$$q = \frac{72\sqrt{3}V}{(h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2)^{3/2}},$$

where V is the volume, and h_1 , h_2 , h_3 , h_4 , h_5 , and h_6 are the edge lengths of the tetrahedron. For a regular tetrahedron $q = 1$. If $q > 0.1$ the mesh quality should not affect the solution's quality. For a hexahedron,

$$q = \frac{24\sqrt{3}V}{\left(\sum_{i=1}^{12} h_i^2\right)^{3/2}},$$

where h_i are the edge lengths. For a cube, $q = 1$. For a prism,

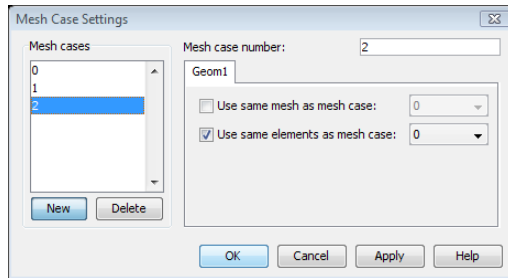
$$q = \frac{36\sqrt{3}V}{\left(\sum_{i=1}^9 h_i^2\right)^{3/2}},$$

where h_i are the edge lengths. Finally, $q = 1$ for a right-angled prism where all edge lengths are equal. In 1D, the element quality and the variable `qual` both always equal 1.

Mesh Cases

Using *mesh cases* you can create meshes with different element sizes and element orders within a single model. This functionality is meant to be used mainly for creating hierarchies for the geometric multigrid preconditioner, but it is also possible to use it to easily switch between different meshes and elements when using another solver.

A mesh case consists of a mesh and corresponding elements that you specify on the **Elements** page in the **Subdomain Settings** dialog box accessed from the **Physics** menu. There is always a current mesh case. All changes made on the **Elements** page and the mesh created in Mesh mode apply to the current mesh case. An indication on the Status bar shows the current mesh case, and you can change the current mesh case from the **Mesh** menu.



When adding a new mesh case, you can specify that it should use the same mesh or the same elements as another mesh case. This means that subsequent changes to the mesh or the elements in that mesh case also apply to the mesh cases that belong to the same group.

The solvers always run using the current mesh case except when you choose the geometric multigrid preconditioner with manual hierarchy specification. Then, in the **Solver Parameters** dialog box, you can explicitly specify which mesh cases to include during the solution process.

Avoiding Inverted Mesh Elements

Inverted Mesh Elements

If you have a mesh that is coarse along a curved boundary you might get problems with inverted mesh elements. This means that a mesh element is wrapped inside-out or has zero area (in 2D) or volume (in 3D). More precisely, there is some coordinate for which the Jacobian matrix for the mapping from local to global coordinates has a negative or zero determinant. In most cases, the linear (straight) mesh elements that you see in a mesh plot are not inverted, but the higher-order curved mesh elements used for computing the solution might be (see “Curved Mesh Elements” on page 484 of the *COMSOL Multiphysics Reference Guide*). Hence, studying the minimum element quality in the **Mesh Statistics** dialog box does not reveal the presence of inverted mesh elements in most cases.

Inverted mesh elements in themselves do not pose any immediate threat to the overall accuracy of your solution. However, if you are using an iterative solver, it might fail to converge. If you reach convergence and the solution looks good, it likely is. It is worth bearing in mind that the faces along which there are inverted elements are less than perfectly resolved. If these faces are important for your results, you may want to pursue a mesh without inverted elements or at least make sure that the mesh resolution is sufficiently fine to guarantee an accurate solution. The easiest way to get an idea of the accuracy is to try a few different meshes and see how the solution changes. If the variation does not exceed your limits of acceptance, you are fine.

The solver prints warnings about inverted mesh elements to the **Log** page in the **Progress** window if they appear. You can avoid problems with inverted mesh elements by reducing the geometry shape order or by modifying the geometry or the mesh.

Using Linear Geometry Shape Order

When solving a model, the solver ensures that no inverted mesh elements are created. This is done by reducing the geometry shape order for the corresponding elements to first order. By default, the solver does this automatically. You can turn off this feature by clearing the **Use first order for inverted elements** check box in the **Model Settings** dialog box, opened from the **Physics** menu. In the corresponding **Geometry shape** area there is also a **Tolerance** edit field that specifies the minimum allowed value of the variable `reldetjacmin` for elements not to be considered as inverted elements. The

default value of this tolerance is 0.1, which means that linear geometry shape order is also used for elements that are very close to being inverted. For more information on the variable `re1detjacmin`, see “Geometric Variables” on page 170.

Alternatively, you can avoid problems with inverted mesh elements by using linear geometry shape order for all elements. You do this by choosing **Linear** from the **Order** list in the **Geometry shape** area in the **Model Settings** dialog box.

Modifying the Geometry or Mesh

If you do not want to use linear geometry shape order to avoid problems with inverted mesh elements you can try any of the following:

- Create an extruded, revolved, or swept 3D mesh instead of using the free mesher.
- Avoid small curved boundaries such as fillets unless they are important for the result.
- Do not use the **Refine Mesh** button on the Main toolbar to create your mesh. Instead choose a finer mesh setting in the list of predefined mesh sizes or give a maximum element size that results in a mesh finer than the original mesh.

Visualizing Inverted Mesh Elements

You can visualize inverted mesh elements both in Mesh mode and in Postprocessing mode. To plot the inverted mesh elements in Mesh mode, follow these steps.

- 1 From the **Mesh** menu, open the **Mesh Visualization Parameters** dialog box.
- 2 Select the **Subdomain elements** check box and clear the **Boundary elements** check box.
- 3 Select the **Element selection** check box and type `re1detjacmin<0` in the **Logical expression for inclusion** edit field.
- 4 Click **OK** to plot the inverted mesh elements and close the dialog box.

If you clear the **Use first order for inverted elements** check box in the **Model Settings** dialog box or if you lower the corresponding tolerance value below zero before solving, you might get inverted mesh elements when solving the problem. To see where these elements are located using the postprocessing tools, follow these steps.

- 1 Open the **Plot Parameters** dialog box. On the **General** page, select the **Subdomain** and **Geometry edges** check boxes.
- 2 Select the **Element selection** check box and type `re1detjacmin<0` in the **Logical expression for inclusion** field.
- 3 Click the **Subdomain** tab and type `re1detjacmin` in the **Expression** field. Click **OK**.

Solving the Model

This chapter lists the various solvers in COMSOL Multiphysics and examines how to choose the best one for a given application. It also reviews which settings to select when solving a wide range of problems of different types and how to analyze the accuracy of the solution and improve convergence. Supplementary information appears in Chapter 5, “Advanced Solver Topics,” of the *COMSOL Multiphysics Reference Guide*.

Selecting a Solver

The COMSOL Multiphysics Solvers

COMSOL Multiphysics includes a number of different solvers for PDE-based problems. Table 6-1 summarizes the available types. They are described in more detail in the next section, “The Solvers and Their Settings” on page 385.

TABLE 6-1: COMSOL MULTIPHYSICS SOLVERS

SOLVER TYPE	USAGE
Stationary	For stationary PDE problems
Time dependent	For time-dependent PDE problems
Eigenvalue	For eigenvalue PDE problems
Parametric	For parameterized sets of stationary PDE problems
Stationary segregated	For stationary multiphysics PDE problems
Parametric segregated	For parameterized sets of stationary multiphysics PDE problems
Time dependent segregated	For time-dependent multiphysics PDE problems
Adaptive	For stationary PDE problems or linear eigenvalue PDE problems using adaptive mesh refinement
Sensitivity	For sensitivity analysis of stationary PDE problems
Optimization [□]	For optimization of stationary PDE problems

[□] This solver type is only available if your license includes the Optimization Lab.

Selecting Analysis Types

For most models, you do not need to select a solver explicitly. Instead, you choose an *analysis type*. The available analysis types depend on the application area; those you can expect to see include stationary, eigenfrequency, transient or time-dependent, and parametric analysis.

You select the analysis type in the Model Navigator. With the exception of the PDE modes and some application modes where there is no choice of analysis types (for example, Electrostatics), the analysis type is an *application mode property* that you can change later on. To do so, choose **Properties** in the **Physics** menu to open the **Application Mode Properties** dialog box or select it from the lists of analysis types in the

Analysis types area in the **Solver Parameters** dialog box (typically one list for each application mode in the model).

The analysis type sets up various equations for the application mode and suggests a default solver. To override this default, open the **Solver Parameters** dialog box and specify the solver of your choice from the **Solver** list. Note, however, that doing so might result in a solver setting that is incompatible with the equations you want to solve (for example, selecting a time-dependent solver directly and not through a selection of analysis type does not change a stationary formulation of an equation into a transient formulation that includes time derivatives). You can disable the automatic solver selection based on the analysis type by clearing the **Auto select solver** check box below the **Analysis types** area in the **Solver Parameters** dialog box.

For multiphysics models, the **Analysis types** area contains lists of available analysis types for all application modes in the model. The *ruling application mode*, which the software indicates using a bold font, suggests a solver based on its analysis type. You can inspect or change the ruling application mode in the **Ruling application mode** list in the Model Navigator. If you change the ruling application mode or its analysis type, the program changes certain solver settings accordingly. It is important that you select appropriate analysis types for all application modes in the model. To change from a stationary to a transient analysis, for example, typically requires that you change the analysis type to a transient (time-dependent) analysis type for all application modes. In some cases, such as thermal stress analysis, you can often treat the structural part as quasi-static and use the quasi-static transient analysis type for the structural application mode (which specifies a time-dependent solver but uses a stationary equation formulation). When you select analysis types in the Model Navigator for the predefined multiphysics couplings, COMSOL Multiphysics uses suitable analysis types for each application mode in the multiphysics coupling.

Beside the analysis type, the set of application mode properties includes the default element type and settings for activating weak constraints. For additional application mode properties, specific to individual application modes, see the *COMSOL Multiphysics Modeling Guide* or the documentation for the module to which the application mode belongs.

Analysis types are not available for PDE modes and for some application modes that only support a single type of analysis, such as the Electrostatics application mode. For the PDE modes, select a solver directly from the **Solver** list. Coefficients which are not relevant for the chosen solver type—such as time derivatives when using the stationary solver—are ignored by the solver.

Selecting a Stationary, a Time-Dependent, or an Eigenvalue Solver

The analysis type generally selects an appropriate solver, so that you do not need to select one yourself. If you nevertheless prefer to make a selection, the first question to ask is whether the problem is stationary or time-dependent.

Most real-world phenomena develop in time, but you might know that the system under study approaches a *steady state* described by a stationary solution.

For a stationary problem, select the stationary solver. When solving the time-dependent coefficient form problem

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

the stationary solver searches for a solution where $\partial u / \partial t = 0$.

The time-dependent solver and the stationary solver both handle linear as well as nonlinear problems.

In some cases you might want to study the natural harmonic oscillations of a time-dependent problem. This involves finding the eigensolutions u with their associated eigenvalues λ in a PDE problem of the following form:

$$\nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + a u = \lambda d_a u - \lambda^2 e_a u$$

Such an analysis is particularly interesting in electromagnetics, structural mechanics, acoustics, and wave propagation. To study the eigensolutions and compute the eigenvalues (or eigenfrequencies), select the eigenvalue solver.

Linear Solvers vs. Nonlinear Solvers

AUTOMATIC NONLINEARITY DETECTION

COMSOL Multiphysics automatically detects nonlinearity, so you normally do not need to decide whether to use a linear or a nonlinear solver.

The automatic detection works through analysis of the variables contributing to the residual Jacobian matrix and the constraint Jacobian matrix. If the algorithm finds that both these matrices are complete and do not depend on the solution, the stationary solver and the parametric solver use the linear solver. Otherwise, they use the nonlinear

solver. Complete here means that the algorithm only found contributing variables for which the correct Jacobian will be computed.

OVERRIDING THE AUTOMATIC NONLINEARITY DETECTION

In some cases, you might want to specify explicitly that the stationary and parametric solvers use the linear solver or the nonlinear solver. Such cases include:

- Linear models where the automatic detection of linearity makes COMSOL Multiphysics use the nonlinear solver. This can happen, for example, for models that involve some less common types of coupling variables (directly or indirectly as part of some boundary conditions). The nonlinear solver usually converges directly for linear problems, but if that is not the case, you can switch to the linear solver.
- Using the linear solver to single-step Newton’s method for a nonlinear problem.
- Using the linear solver to solve a linearized (nonlinear) problem.

To explicitly control which solver algorithm the stationary and parametric solvers use, click the **Stationary** tab in the **Solver Parameters** dialog box. Then, in the **Linearity** list, select **Linear** to use the linear solver or **Nonlinear** to use the nonlinear solver. With the default setting, **Automatic**, COMSOL Multiphysics detects nonlinearity automatically.

WHICH MODELS ARE NONLINEAR?

How do you determine if a problem is linear or nonlinear? Finding out is not always easy, in particular because some physics modes hide their underlying equations.

For most of the standard application modes you can apply the following criterion: if any coefficient or material property contains a dependent variable the model is nonlinear. The same holds true for models based on a PDE in the coefficient form, again with the same criterion.

Note a few special cases that arise with some application modes. First, in the Heat Transfer by Conduction application mode, if either of the coefficients C_{trans} or C (for radiation terms that depend on temperature according to the Stefan-Boltzmann law) is nonzero, the problem is nonlinear. Second, the Incompressible Navier-Stokes application mode is always nonlinear unless $\rho = 0$, resulting in the linear Stokes equations.

Whether your problem is linear or nonlinear, it is important to realize that the solvers break it down into one or several linear systems of equations. The choice of solver for linear systems thus affects the solution time and memory requirements also for nonlinear models. The section “Linear System Solver Selection Guidelines” on page 428 advises on how to make this choice.

THE IMPORTANCE OF A CORRECT JACOBIAN MATRIX

As mentioned above, the solvers break down each problem—whether linear or nonlinear—into one or several linear systems of equations by approximating the given problem with a linearized problem. The coefficient matrix of the discretized linearized problem is called the *Jacobian matrix* (or *stiffness matrix*). In most cases COMSOL Multiphysics computes a correct Jacobian matrix, especially when using the weak solution form. For other cases, though, the software treats certain terms in the PDEs and boundary conditions as if they did not depend on the sought solution, and then an incorrect Jacobian results.

The consequences of an incorrect Jacobian matrix depend on the solver in use:

- The linear stationary solver and the eigenvalue solver simply give an incorrect solution.
- The nonlinear stationary solver and the time-dependent solver take longer time to convergence to the correct solution, and in some cases the solver even fails to find a solution. However, if the ignored terms have a very weak dependence on the sought solution, the impact on convergence speed is small.

An incorrect Jacobian matrix can occur in the following cases (see “The Linear or Linearized Model” on page 386):

- If you supply an incorrect derivative of some of the functions in the **Functions** dialog box, and the software uses this function in some PDE or boundary condition with arguments that depend on the solution.
- If you use a COMSOL Script or MATLAB M-file function for which you have specified no derivative, and then call this function in some PDE or boundary condition with arguments that depend on the solution. To avoid this problem, define the function and its derivatives in the **Functions** dialog box (see “Using Functions Based on Interpolated Data” on page 154).
- If you use nonanalytic functions in a complex-valued problem, such as `real`, `imag`, `conj`, `abs`, or `angle`.
- If you use the `nojac` operator.
- If you use a solution form that is not suited for the equations you are solving. Selecting **Automatic** in the **Solution form** list on the **Advanced** page in the **Solver Parameters** dialog box ensures a correct solution form (this is the default setting).

See the section “Solution Form” on page 526 of the *COMSOL Multiphysics Reference Guide* for additional information.

WORKING WITH COMPLEX-VALUED PROBLEMS

When a problem contains complex numbers, be sure to consider the following two aspects:

- For time-dependent problems, the time-stepping algorithm must know if a problem is complex valued. If your model uses a complex-valued initial solution, COMSOL Multiphysics detects this and classifies the problem accordingly. If the problem is complex-valued because of a complex-valued PDE coefficient or other material property, go to the **Solver Parameters** dialog box, find the **General** page, and select the **Allow complex numbers** check box.
- If you expect to receive complex outputs from real inputs in elementary functions such as `sqrt`, `log`, and `asin`, then go to the **Solver Parameters** dialog box, click the **Advanced** tab, and select the **Use complex functions with real input** check box. By default, COMSOL Multiphysics gives an error message if a real input to a function generates a complex output, for instance `sqrt(-1)`.

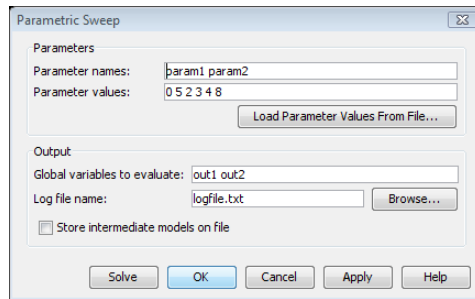
Setting Up Parametric Sweeps

In the **Parametric Sweep** dialog box, available from the **Solve** menu, you can set up a single solver call that performs several runs using the currently selected solver while varying a set of model parameters. This feature is useful, for example, if you want to solve a parameterized set of the transient PDE problems. Another situation when to use a parametric sweep, is when you have a very large parameter set that would result in an impractically large model MPH-file; with a parametric sweep, you can instruct COMSOL Multiphysics to store the results in a number of separate files instead.

If you are running COMSOL Multiphysics in distributed mode (see “Distributed-Memory Parallel COMSOL” on page 74 of the *COMSOL Installation and Operations Guide*), the software distributes the parameter sweep on the nodes; it then assumes that the parameters are independent in the sense that the program can split the list in any way without causing convergence problems for each separate problem.

Permissible parameters are those that appear in the model where a constant could have been used instead, for example, parameters specifying values in the dialog boxes on the **Physics** menu. Specify the parameters to vary in the **Parameter names** edit field. When solving for several parameters, enter the list as a vector where the parameter sets are stored consecutively. For example, if there are two parameters, the entry `0 5 2 3`

4 8 in the **Parameter values** edit field means that the parametric sweep is performed with the parameter-value pairs (0, 5), (2, 3), and finally (4, 8).



The Parametric Sweep dialog box.

You can also use the **Load Parameter Values From File** option, which loads the values from a text file into the **Parameter values** edit field. In this text file, the parameter sets must be listed row-wise with the values in each set separated by a space or tab.

In the **Global variables to evaluate** edit field, located in the **Output** area, you can specify a list of variables that COMSOL Multiphysics evaluates and stores in the log file for each separate simulation in the sweep. Enter a file name in the **Log file name** edit field. COMSOL Multiphysics saves a file with the specified name containing the parameter values and the corresponding evaluated variables in the current working directory. Click the **Browse** button to specify a different location.

If you want to store the full model results for each intermediate simulation in an MPH-file, select the **Store intermediate models on file** check box.

Setting Up Geometric Parametric Sweeps

If you have licenses for the CAD Import Module and SolidWorks or Autodesk Inventor, you can set up automatic parametric sweep also for varying the model geometry. To do this, use the **Geometric Parametric Sweep** dialog box available from the **SolidWorks Connection** or **Inventor Connection** submenu in the **File** menu.

The parameters you specify in this dialog box, which is identical in design to the **Parametric Sweep** dialog box, can also be used to update the model in the CAD software that you are currently connected to. The parameters must correspond to parameters already defined in the CAD model that is open and active in the CAD software. Both model-defined and user-defined parameters in Autodesk Inventor are supported, while only user-defined parameters in SolidWorks are supported.

The Solvers and Their Settings

The Stationary Solver

For *stationary problems*, also called *static problems* or *steady-state problems*, use the stationary solver. See “Selecting a Stationary, a Time-Dependent, or an Eigenvalue Solver” on page 380 to decide if the stationary solver should work for a given model.

THE STATIONARY MODEL

The stationary solver works on linear and nonlinear stationary PDE problems, for example, the coefficient-form PDE

$$\nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

In a nonlinear model, one or more of the coefficients c , α , γ , β , a , or f (or any of the boundary coefficients q , g , h , or r) typically depend on the solution u or its spatial derivatives (see “Which Models are Nonlinear?” on page 381).

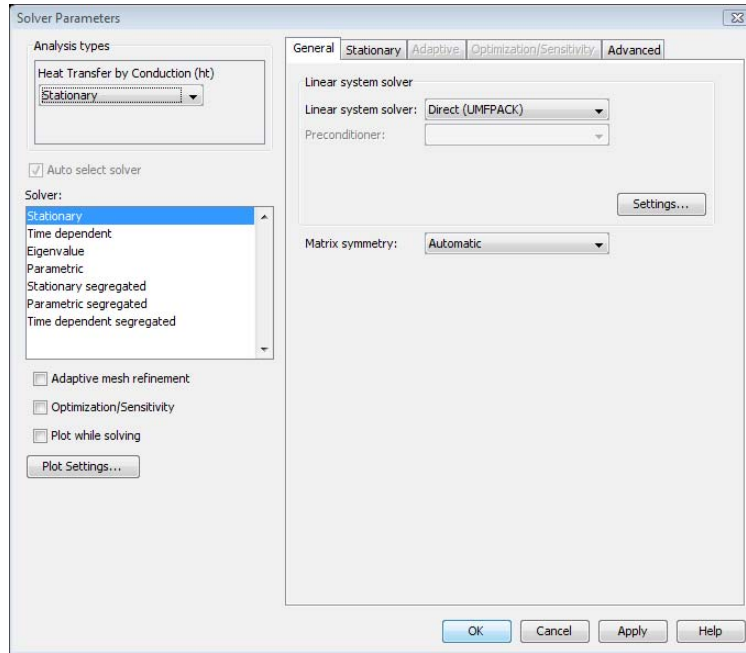
By default, COMSOL Multiphysics solves both linear and nonlinear problems using the weak solution form. This means that the entire model is converted before solution to a form which can symbolically be written very compactly as

$$\begin{aligned} L(u) + (\nabla r_f)^T \mu &= 0 \\ r(u) &= 0 \end{aligned} \tag{6-1}$$

where the *residual*, $L(u)$, is the sum of all weak form contributions in the model, $r(u)$ is a column vector with all constraints and ∇r_f is the Jacobian of the corresponding constraint force expressions (usually, all constraints in the model are ideal, in which case $r_f(u) = r(u)$). The goal of the stationary solver is to find a solution u which satisfies Equation 6-1.

USING THE STATIONARY SOLVER

In the **Solver Parameters** dialog box, select **Stationary** from the **Solver** list.



The General page settings in the Solver Parameters dialog box.

The choice of analysis type sets up a default for the linear system solver. The section “Selecting a Linear System Solver” on page 426 describes how to override the default settings. For the iterative linear solvers you can also select a preconditioner.

The stationary solver normally chooses a linear or nonlinear solver depending on the linearity of the model, but it is possible to override the automatic selection (see “Linear Solvers vs. Nonlinear Solvers” on page 380).

THE LINEAR OR LINEARIZED MODEL

If a model is nonlinear, the linear solver solves the corresponding *linearized* model, evaluated at a specified *linearization point*. The result is equivalent to adding a Newton correction to the linearization point. Repeated linear solutions, updating the linearization each time, is therefore equivalent to an undamped Newton method and will often converge rapidly towards the solution of the nonlinear model.

The linearization point is a given solution that you specify in the **Solver Manager**. As the linearization point you can use the initial value, the current solution, the stored solution, or zero (see “The Solver Manager and Solver Scripting” on page 436). To solve a nonlinear problem through repeated calls to the linear solver, use the **Current solution** setting, or click the **Restart** button on the Main toolbar. If the model is in fact linear, the linear solver will always provide the correct solution in one step, independently of the linearization point.

When using the default weak solution form, the linearized model corresponding to Equation 6-1 is

$$\begin{aligned} \nabla L(u_0)(u - u_0) - (\nabla r_f(u_0))^T \mu &= L(u_0) \\ \nabla r(u_0)(u - u_0) &= r(u_0) \end{aligned} \quad (6-2)$$

where the quantities $\nabla L(u_0)$ and $\nabla r(u_0)$ are known as the Jacobian of the problem and the constraint Jacobian, respectively. The right-hand side consists of the residual $L(u_0)$ and constraint residual $r(u_0)$. The weak solution form calculates these quantities automatically from the weak expressions and evaluates them at the current linearization point, u_0 .

If the solution form is the default weak form, COMSOL Multiphysics generates a complete linearization, taking care of all dependencies including coupling variables and derivatives in boundary conditions. However, an incomplete linearization can occur if you fail to provide correct derivatives of functions that you have defined.

If the software solves the model in the general form (by choosing the general solution form), the linearized model is

$$\begin{cases} \nabla \cdot (-c \nabla \tilde{u} - \alpha \tilde{u} + \gamma) + \beta \cdot \nabla \tilde{u} + a \tilde{u} = f & \text{in } \Omega \\ \mathbf{n} \cdot (c \nabla \tilde{u} + \alpha \tilde{u} - \gamma) + q \tilde{u} = g - h^T \mu & \text{on } \partial\Omega \\ h \tilde{u} = r & \text{on } \partial\Omega \end{cases}$$

where $\tilde{u} = u - u_0$, and u_0 is the linearization point. The coefficients

$$\begin{array}{lllll} c = \frac{\partial \Gamma}{\partial \nabla u} & \alpha = \frac{\partial \Gamma}{\partial u} & \gamma = \Gamma & \beta = \frac{\partial F}{\partial \nabla u} & a = \frac{\partial F}{\partial u} \\ f = F & q = \frac{\partial G}{\partial u} & g = G & h = \frac{\partial R}{\partial u} & r = R \end{array}$$

are evaluated for $u = u_0$. The meaning of these derivatives becomes clearer with indices:

$$\begin{aligned}
 c_{lkji} &= -\frac{\partial \Gamma_{lj}}{\partial \left(\frac{\partial u_k}{\partial x_i} \right)} & \alpha_{lkj} &= -\frac{\partial \Gamma_{lj}}{\partial u_k} & \beta_{lkj} &= -\frac{\partial F_l}{\partial \left(\frac{\partial u_k}{\partial x_j} \right)} \\
 a_{lk} &= -\frac{\partial F_l}{\partial u_k} & q_{lk} &= -\frac{\partial G_l}{\partial u_k} & h_{mk} &= -\frac{\partial R_m}{\partial u_k}
 \end{aligned}$$

Note that this linearization only accounts for the most common ways that the PDE and boundary conditions can depend on the solution. For instance, it does not account for occurrences of derivatives in boundary conditions and coupling variables.

If the solution form is coefficient form, COMSOL Multiphysics uses a simple linearization, namely

$$\begin{cases}
 \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f & \text{in } \Omega \\
 \mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) + q u = g - h^T \mu & \text{on } \partial \Omega \\
 h u = r & \text{on } \partial \Omega
 \end{cases}$$

where the software evaluates the coefficients $d_a, e_a, c, \alpha, \gamma, \beta, a, f, q, g, h$, and r for the given linearization point $u = u_0$. This linearization does not account for the dependence of the coefficients on the solution u .

For more information on linearization, see “The Importance of a Correct Jacobian Matrix” on page 382.

THE DISCRETIZED LINEARIZED MODEL

After finite-element discretization, you can write a linear or linearized model as

$$\begin{aligned}
 K(U_0)(U - U_0) + N_F(U_0)\Lambda &= L(U_0) \\
 N(U_0)(U - U_0) &= M(U_0)
 \end{aligned}$$

where U_0 is the solution vector corresponding to the linearization point u_0 , U is the solution vector (corresponding to u), and Λ is the Lagrange multiplier vector (corresponding to μ). K is the Jacobian (stiffness) matrix, L is the equation residual vector, N is the constraint Jacobian matrix, N_F is the constraint force Jacobian, and M is the constraint residual vector. Note the similarity and term-by-term correspondence to the linearized weak form in Equation 6-2.

If your model has only ideal constraints then

$$N_F = N^T.$$

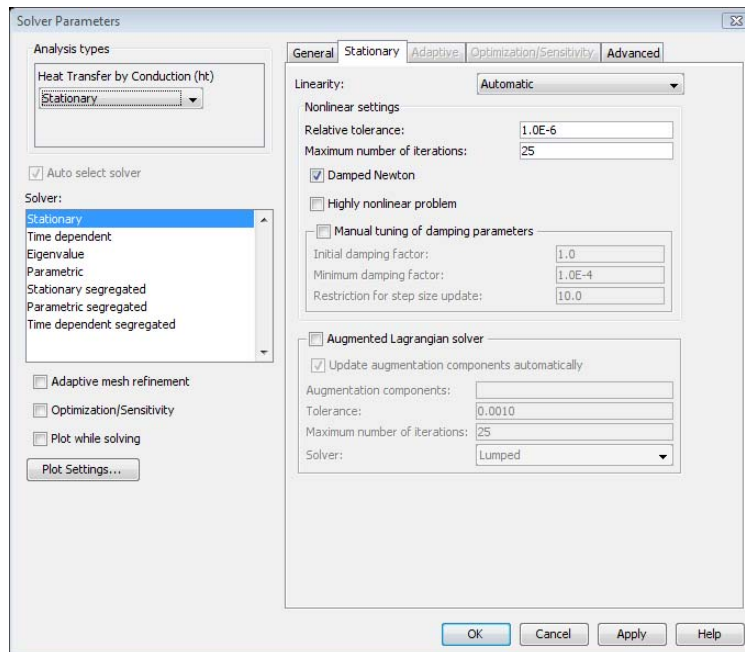
If the model is linear, K , N_F , and N are independent of U_0 ; setting $U_0 = 0$, you can write

$$\begin{bmatrix} K & N_F \\ N & 0 \end{bmatrix} \begin{bmatrix} U \\ \Lambda \end{bmatrix} = \begin{bmatrix} L(0) \\ M(0) \end{bmatrix}.$$

Before applying the linear system solver, COMSOL Multiphysics transforms this system using a constraint-handling method (see “Constraint Handling” on page 531 of the *COMSOL Multiphysics Reference Guide* for details).

NONLINEAR SOLVER SETTINGS

To specify settings for the nonlinear solvers, click the **Stationary** tab. It is rarely necessary to modify these settings. For information about the options in the **Linearity** list, see “Linear Solvers vs. Nonlinear Solvers” on page 380.



Settings for the nonlinear solver.

Nonlinear Settings

The value in the **Relative tolerance** edit field (default value 10^{-6}) applies to a convergence criterion based on a weighted Euclidean norm for the estimated relative error; the nonlinear-solver iterations stop when the relative error is less than the relative tolerance. For details on the convergence criterion and the nonlinear solver algorithm, see “The Nonlinear Solver Algorithm” on page 533 of the *COMSOL Multiphysics Reference Guide*.

The value in the **Maximum number of iterations** edit field limits the number of nonlinear iterations (default = 25).

Normally, the solver uses a damped Newton method. If you want the solver to always attempt to take full Newton steps, clear the **Damped Newton** check box.

If the problem is highly nonlinear, selecting the **Highly nonlinear problem** check box can help make the solver more careful. Try this option if the solution does not converge.

Manual Tuning of Damping

To tune the damping manually, select the **Manual tuning of damping parameters** check box.

The **Minimum damping factor** is a safeguard against perpetual Newton iterations. When the damping factor drops below the level specified in this edit field, the solver stops and issues the following error message: **Failed to find a solution: No convergence, even when using the minimum damping factor. Returned solution has not converged**. The default value is 10^{-4} if the **Highly nonlinear problem** check box is cleared, otherwise it is 10^{-8} .

The default value in the **Initial damping factor** edit field is 1 if the **Highly nonlinear problem** check box is cleared or 10^{-4} if it is selected. The parameter in the **Restriction for step size update** edit field (default = 10) sets a restriction for the damping-factor update in the Newton iteration. Each time the solver updates the damping factor, it can change up or down by at most this factor.

The Augmented Lagrangian Solver

If you select the **Augmented Lagrangian solver** check box, the solving procedure is split into two substeps that the solver repeats until a convergence criterion is met or until the maximum number of iterations is reached. You control the splitting of the problem into substeps using the **Augmentation components** edit fields. The solver first computes the solution for the main components, which are the solution components that are not augmentation components, while keeping the augmentation components fixed. Then it solves for the augmentation components while keeping the main components fixed.

This second solution step uses a linear solution approach. Therefore, with the main components fixed, the solver algorithm assumes that the augmentation components satisfy a linear equation. If you select the **Update augmented components automatically** check box, the application modes fill in these component names automatically.

The Time-Dependent Solver

Use the time-dependent solver to find the solution to linear and nonlinear *time-dependent PDE problems*, also known as *dynamic problems* or *unsteady problems*. See the section “Selecting a Stationary, a Time-Dependent, or an Eigenvalue Solver” on page 380 to decide if the time-dependent solver should work for a given model.

THE TIME-DEPENDENT PROBLEM

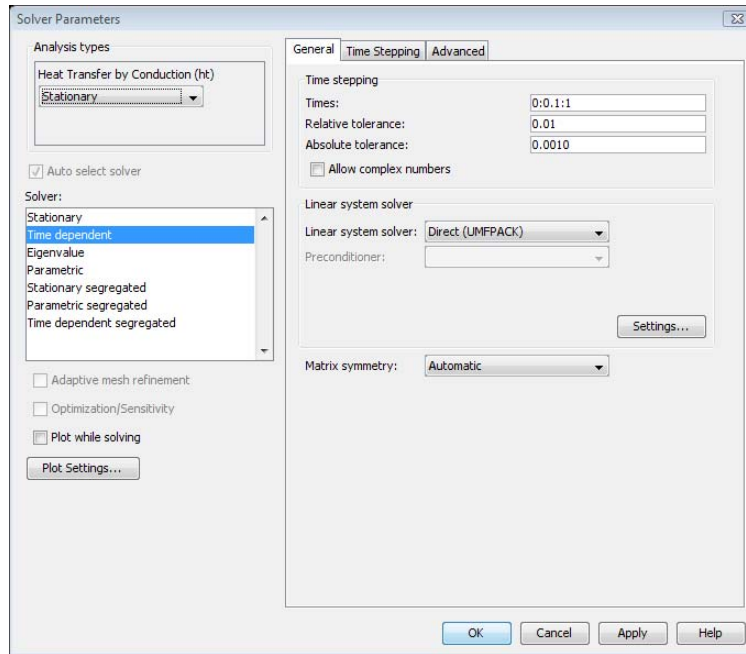
You can express a time-dependent PDE in coefficient form as

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

on the computational domain in 1D, 2D, or 3D. The PDE together with boundary conditions and initial conditions fully define the problem. Many application modes support time-dependent (transient) modeling using a time-dependent PDE formulation.

USING THE TIME-DEPENDENT SOLVER

To use the time-dependent solver, go to the **Solver Parameters** dialog box, find the **Solver** list, and select **Time dependent**.



Selecting the time-dependent solver and specifying output times.

Specifying Output Times

In the **Time stepping** area enter the output times in the **Times** edit field in the form of a vector of times at which you want the solution to the time-dependent model (for another option see “Setting the Time Steps for the Output and the Solver” on page 395). The syntax `0:0.1:1` represents a vector of times starting at 0 with steps of 0.1 up to 1. The relevant time span depends on the model’s dynamics.

Setting Tolerances

It is sometimes essential for accuracy or performance that you set absolute and relative tolerance parameters for the time-dependent solver. In the **Relative tolerance** edit field you enter a positive number (default = 0.01). In the **Absolute tolerance** edit field you enter a single positive number (default = 0.001) or a space-separated list whose entries alternate between degree-of-freedom names (names of the dependent variables) and positive scalars. For example, to use an absolute tolerance of 0.001 for the velocity

components u and v and an infinite absolute tolerance for the pressure p in a 2D fluid-flow model, type the following space-separated pairs of names and corresponding tolerance values: `u 0.001 v 0.001 p inf`. It is also possible to use wild cards in the space-separated list that are matched against the available degree-of-freedom names. For example, second-order vector elements have the following degree-of-freedom names: `tAxAyAz10`, `tAxAyAz20`, and `tAxAyAz21`. Type the following to set the absolute tolerance to all of them to the same value: `tA* 1e-7`. The wild card match also allows `?`, which represents any single character. The absolute tolerance values have the unit of the corresponding solution component in the model's unit system. For example, if the model uses the MPa unit system, the displacement component's absolute tolerance values have the unit millimeter.

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. The step is accepted if

$$\left(\frac{1}{N} \sum_i \left(\frac{|E_i|}{A_i + R|U_i|} \right)^2 \right)^{1/2} < 1$$

where A_i is the absolute tolerance for DOF i , R is the relative tolerance, and N is the number of degrees of freedom. The accumulated (global) error can be larger than the sum of the local errors for all the integration steps. However, the solver's error estimate is often too pessimistic, which means that the estimated local error typically is of the same order of magnitude as the true global error.

Note: 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.

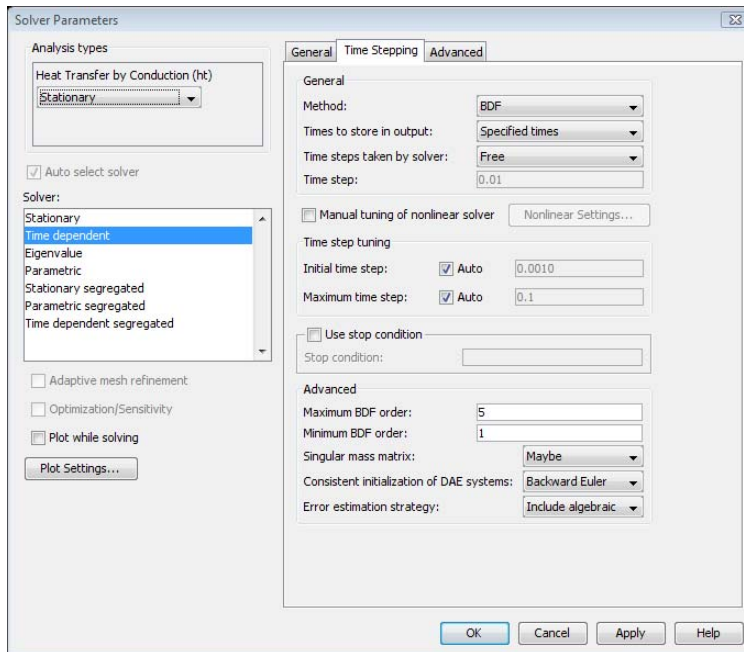
Note: For DAEs (differential-algebraic equations), you can exclude the algebraic equations from the error estimation. See “Settings for Differential-Algebraic Equations” on page 400 for details.

Setting Other Solver Properties

- The selection of analysis type sets up defaults for the linear system solver. The section “Selecting a Linear System Solver” on page 426 describes how to override the default settings.
- If the model contains complex numbers, select the **Allow complex numbers** check box (see “Working with Complex-Valued Problems” on page 383).
- For symmetric matrices, use the criteria listed in the section “Which Problems Are Symmetric?” on page 431.
- If the transient analysis contains a large number of time steps in the results data (output times), click the **Advanced** tab and select the **Store solution on file** check box. For more information about this option, see “Storing Solutions on File” on page 526 of the *COMSOL Multiphysics Reference Guide*.

SETTINGS FOR THE TIME-DEPENDENT SOLVER

To specify additional settings for the time-dependent solver, click the **Time Stepping** tab.



Settings for the time-dependent solver.

Selecting Integration Method

There are two methods available in COMSOL Multiphysics for solving time-dependent problems: the *backward differentiation formula* (BDF) and the *generalized- α* method. You select which method to use by choosing **BDF** or **Generalized alpha** from the **Method** list. BDF is a robust method that is commonly used for a wide range of problems. Generalized- α was first derived for problems in structural mechanics but has become a popular method also in other application areas. For more information about the two methods, see “The Time-Dependent Solver Algorithm” on page 535 of the *COMSOL Multiphysics Reference Guide*. The application modes in COMSOL Multiphysics and the modules use a suitable default integration method for solving the time-dependent equation formulations that they support.

Setting the Time Steps for the Output and the Solver

The internal time steps in the solver are more or less independent of the output times you specify for postprocessing (by default). To obtain the solution results at the internal time steps instead of the times in the **Times** edit field, go to the **Times to store in output** list and select **Time steps from solver**. To view the time steps the solver takes, click the **Log** tab in the **Progress** window.

To control how the solver takes its internal time steps, select the appropriate option in the **Time steps taken by solver** list. For the **Free** option (the default) the solver chooses its time steps arbitrarily; in other words, it ignores the **Times** list when selecting them. For the **Strict** option, the solver must take a time step at each entry in the time list and it adds time steps in between as necessary. Use this feature when an equation involves impulses or irregularities that the time stepping must not skip over. Adding times around each impulse ensures that the solver can resolve them. For the **Intermediate** option, the solver must take at least one time step in each subinterval of the time list. This option is similar to **Strict** except it gives the solver somewhat more freedom and therefore is more efficient. If you choose the option **Initialization only**, the solver only computes consistent initial values of your problem and then immediately stops.

In the area **Time step tuning**, you can specify additional properties which influence the time steps the solver takes. Clearing the **Auto** check box for **Initial time step** makes it possible to set the initial internal time step. Similarly, an upper limit on the internal time steps can be specified when the **Auto** check box for **Maximum time step** is cleared. When solving problems of wave type, the nature of the solution can sometimes lead to an oscillating error estimate, which in turn results in the time step oscillating in an inefficient manner. When the method generalized- α is used, selecting the check box **Use time step increase delay (steps)** can prevent such oscillating time steps by making the solver more restrictive in increasing the time step. A counter keeps track of the

number of consecutive time steps for which a time step increase has been warranted. When this counter exceeds the number entered in the edit field (default is 15), the time step is increased and the counter is reset to zero. So, a large integer in the edit field makes the solver more restrictive in increasing the time step and the value zero in the edit field is equivalent to disabling the delay mechanism (clearing the check box).

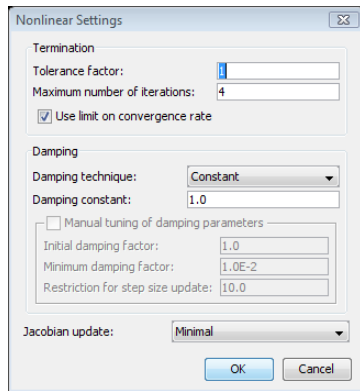
In some cases, the solver is too pessimistic or inefficient when choosing the time step. When you use the generalized- α method, the automatic choice of time step can be disabled by choosing **Manual** from the **Time steps taken by solver** list. Specify the desired time step in the **Time step** edit field as a scalar, as a vector of times the solver should solve at, or as an expression containing variables which are *global expressions* (see “Expression Variables” on page 181).

Note: When using manual time stepping, there is no estimation of the local error made during the time step (see “Setting Tolerances” on page 392). However, the absolute and relative tolerances are still important because they influence the accuracy to which the nonlinear system is solved (see “Manual Tuning of the Nonlinear Solver” on page 396). Therefore, set the tolerances to the desired accuracy in the solution.

Manual Tuning of the Nonlinear Solver

In each time step, COMSOL Multiphysics solves a nonlinear system of equations using a damped Newton method (or a linear system, if the PDE problem is linear). You can tune the solution process manually in the dialog box that opens when you click the

Nonlinear Settings button (when using the BDF time-stepping method, first select the **Manual tuning of nonlinear solver** check box to activate this button).



Dialog box for manual tuning of the nonlinear solver.

The value in the **Tolerance factor** edit field (default value 1) controls the convergence criterion for the nonlinear solver. The convergence criterion is based on a weighted norm, including the relative and absolute tolerances, of the estimated error in the solution. For convergence, the estimated error should be smaller than a tolerance which has been chosen such that other errors are likely to dominate. This tolerance is multiplied with the **Tolerance factor**; a value less than one forces the nonlinear solver to solve more accurately, while a value larger than one allows the nonlinear solver to terminate at a larger estimated error.

The value in the **Maximum number of iterations** edit field limits the number of nonlinear iterations (default is 4).

During the Newton iterations, the nonlinear solver estimates a linear convergence rate. If the convergence is too slow, the solver terminates and updates the Jacobian or reduces the time step, because it is not likely that the solver reaches the desired tolerance before the maximum number of iterations has been reached. In some cases, an initial apparent divergence of the Newton iterations is followed by convergence. For such problems, it might be necessary to disable the termination at poor convergence by clearing the **Use limit on convergence rate** check box.

The damping factor used in the damped Newton method can be controlled in the **Damping** area. From the **Damping technique** list, you can choose between **Constant** and **Automatic** damping factor. If you select **Constant**, enter the damping factor to be used in the **Damping constant** edit field. When **Automatic** is selected, the solver determines

an appropriate damping factor for each Newton iteration. With **Manual tuning of damping parameters** selected, you can influence the automatic choice of damping factor (see “Manual Tuning of Damping” on page 390).

When **Constant** is chosen, you can control when to update the Jacobian matrix in the **Jacobian update** list. The default choice is **Minimal**, in which case the Jacobian is updated as seldom as possible; a new Jacobian is computed only if the nonlinear solver has failed to converge with an old Jacobian or if the time step (or order of the method when the BDF method is used) has changed sufficiently much. If you select **Once per time step**, the Jacobian is updated in the beginning of each time step. This can be efficient for problems that usually require an updated Jacobian for convergence. For even more difficult problems, selecting **On every iteration** to update the Jacobian as often as possible might be the most efficient choice.

Specifying a Stop Condition

You can force the solver to stop based on a stop condition. To do so, select the **Use stop condition** check box. The solver then stops before the expression in the **Stop condition** edit field becomes negative (the solver does not try to resolve the exact time for the zero crossing). The expression must be a scalar value that has no variation in space, typically a coupling variable with a global destination. For example, if you have defined an integration coupling variable $utot$ that is the integral of the solution on the domain, typing $0.25 - utot$ in the **Stop condition** edit field forces the solver to stop at the last time step before $utot$ becomes larger than 0.25.

When using a deformed mesh with remeshing, the Moving Mesh (ALE) application mode provides a default expression that is the minimum quality of the deformed mesh minus 0.05 (see “Stop Condition” on page 468 of the *COMSOL Multiphysics Modeling Guide*).

Advanced Properties for the Time-Stepping Algorithm

In the **Advanced** area you control additional settings for the time-stepping algorithm.

Order of the Time-Stepping Method When you use the BDF time-stepping method, you can control the degree of the interpolating polynomial used in the **Maximum BDF order** edit field. Supply an integer between 1 and 5 (the default). A smaller value results in a more stable numerical scheme. In wave-equation type problems, for example, decreasing the order to 2 often gives better results because the higher-order algorithms are only stable when the time step is small, while the order-2 algorithm is unconditionally stable. On the other hand, the low-order algorithms have more numerical damping than the higher-order algorithms. You can reduce numerical

damping by requiring the solver to take smaller time steps. The generalized- α method is often a better time-stepping method for wave-equation type problems.

For some wave-equation type problems the solver decreases the order of the BDF method at certain points of the simulation to an unnecessarily low order (typically from order 2 to order 1). In the **Minimum BDF order** edit field you can specify the lowest order the solver is allowed to use. Note that this minimum order does not apply to the start-up phase of the simulation, where the solver still can use any order.

Damping of high frequencies When you use the generalized- α method, you control how much damping of high frequencies that the solver provides in a time step by entering a value between zero and one in the **Amp. factor for high frequencies** edit field. For a linear problem, the algorithm multiplies the amplitude of the highest possible frequency by this factor each time step. Hence, choosing zero (for a linear problem) annihilates the highest frequency in one time step and choosing one (for a linear problem) preserves the amplitude of the highest frequency (and of all other frequencies as well). Although this precise interpretation of the value in the **Amp. factor for high frequencies** edit field does not hold for nonlinear problems, a value close to zero means that high frequencies are efficiently damped and a value close to one means little damping.

Note: Damping also affects low and moderate frequencies. With too much damping, frequencies in the solution that are relevant and of interest might be lost. The best choice of the value in the **Amp. factor for high frequencies** edit field is highly problem dependent.

Prediction of the solution Solving a nonlinear system of equations (for the solution at the next time step) using Newton's method requires an initial guess of the solution. When using the generalized- α method, you can specify in the **Predictor** list how the solver derives the initial guess. With the default value **Linear**, a guess is constructed through linear extrapolation using the current solution, the current time derivative, and the time step. This is usually the preferred predictor. In some cases, linear extrapolation can give a poor initial guess, even leading to failure of convergence for the nonlinear solver. For such problems, it can be more efficient to select **Constant** from the **Predictor** list, in which case the initial guess for the solution at the next time step is the current solution.

Settings for Differential-Algebraic Equations By default, the solver determines whether a system is *differential algebraic* (see “Differential-Algebraic Problems” on page 400) by looking for zero-filled rows or columns in the mass matrix (this corresponds to **Maybe** in the **Singular mass matrix** list). If the model includes a differential-algebraic equation (DAE) where the mass matrix has no zero-filled rows or columns, select **Yes** in the **Singular mass matrix** list.

A system of DAEs implies some constraints on the initial values, for instance, that a stationary (algebraic) equation must be satisfied. By default the solver perturbs the given initial values so that they become consistent with these constraints. For instance, you need not specify the correct initial value for the pressure in a model using the incompressible Navier-Stokes equations; the solver computes it automatically. The selection in the **Consistent initialization of DAE systems** list controls how the solver perturbs the initial values:

- Select **Backward Euler** (the default) for index-1 and index-2 DAEs. The solver perturbs the initial values of all DOFs by taking a backward Euler step, giving a small perturbation to the differential DOFs.
- Select **On** for index-1 DAEs (this option is only available for the BDF time-stepping method and with the **Manual tuning of nonlinear solver** check box cleared). The solver fixes the values of the differential DOFs (to those given by the initial value), and then it solves for the initial values of the algebraic DOFs and the time derivative of the differential DOFs. When applicable, this choice is preferable to **Backward Euler** because it does not perturb the values of the differential DOFs.
- Select **Off** to indicate that the initial values already are consistent, which means that the solver does not modify them. This option is seldom useful.

For a DAE system, selecting **Exclude algebraic** from the **Error estimation strategy** list excludes the algebraic DOFs from the error norm when computing the time discretization error. For instance, the accuracy in the pressure in a Navier-Stokes model is sometimes not important. Using this option often results in a quicker time stepping.

Manual Control of Reassembly If your model uses periodic boundary conditions, identity conditions, or coupling variables, you might be able to speed up the solving process by doing some manual tuning of the reassembly options; see “Manual Control of Reassembly” on page 528 of the *COMSOL Multiphysics Reference Guide* for more information.

DIFFERENTIAL-ALGEBRAIC PROBLEMS

The system

$$\begin{aligned}\frac{\partial u}{\partial t} &= v \\ u^2 + v^2 &= 1\end{aligned}\tag{6-3}$$

is an example of a *differential-algebraic system* (DAE). The first equation is a differential equation, while the second is an algebraic relationship. You can derive a corresponding ODE system by differentiating the second equation with respect to t :

$$2u\frac{\partial u}{\partial t} + 2v\frac{\partial v}{\partial t} = 0.\tag{6-4}$$

Substituting the first equation in Equation 6-3 into Equation 6-4 leads to

$$\frac{\partial v}{\partial t} = -u,$$

which together with the first equation forms a system of ODEs. The variable v is of *index 1* because it is necessary to differentiate the second equation one time to get an expression for the derivative of v . The variable u is of *index 0* because you already have an equation for its derivative from the start.

If a system of PDEs contains both dynamic PDEs (with time derivatives) and stationary PDEs (without time derivatives), the corresponding space-discretized system is a DAE system. In other words, the mass matrix is singular because it contains zero-filled rows. For instance, the incompressible Navier-Stokes equations

$$\begin{aligned}\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - \Delta u + \frac{\partial p}{\partial x} &= 0 \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} - \Delta v + \frac{\partial p}{\partial y} &= 0 \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0\end{aligned}$$

give rise to a DAE system when discretizing the space because the third equation turns into an algebraic relationship. Here, the pressure p is of index 2 because you must differentiate the third equation twice with respect t , and then you must differentiate the first and second equations once to deduce an expression for the time derivative of p . The velocities u and v are of index 0.

Index-0 variables are also called *differential variables*, and other variables are called *algebraic variables*. Models with DAE discretizations are often more difficult to solve than those with ODE discretizations. For additional information, see Ref. 1, and for

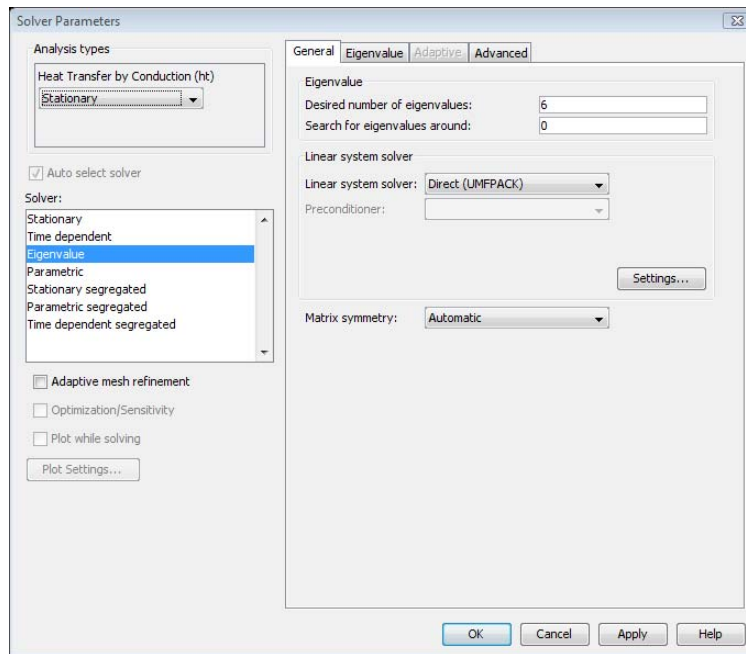
information about solver settings for DAEs see “Settings for Differential-Algebraic Equations” on page 400.

The Eigenvalue Solver

The eigenvalue solver handles linear or linearized *eigenvalue problems*, which in some cases alternatively are called *eigenfrequency problems*. The solver’s output consists of the eigenvalues (or eigenfrequencies) and their associated eigenmodes.

USING THE EIGENVALUE SOLVER

To use this solver, go to the **Solver Parameters** dialog box, find the **Solver** list, and select **Eigenvalue**.



Selecting the eigenvalue solver.

Specifying the Output

Enter the number of eigenvalues COMSOL Multiphysics should compute in the **Desired number of eigenvalues** edit field (default = 6 eigenvalues).

Specify where to look for the desired eigenvalues in the **Search for eigenvalues around** edit field. This value is often called the *shift*. Enter a real or complex scalar; the default

value is 0, meaning that the solver tries to find eigenvalues close to 0. Often, but not always, the solver finds the desired number of eigenvalues that are closest to the shift. For example, for the default settings the solver tries to find the 6 smallest eigenvalues. Occasionally, the solver instead finds the 5 smallest eigenvalues and the seventh smallest eigenvalue. To find all the 6 smallest eigenvalues, increase the desired number of eigenvalues to 8, for instance.

Note: The eigenvalue algorithm does not work reliably when the value in the **Search for eigenvalues around** edit field is equal to an eigenvalue.

Setting General Solver Properties

- The selection of analysis type sets up a default for the linear system solver. The section “Selecting a Linear System Solver” on page 426 describes how to override the default settings.
- For symmetric matrices, use the criteria given in the section “Which Problems Are Symmetric?” on page 431.
- If the eigenvalue model is obtained as the linearization of a nonlinear model, specify the linearization point in the **Solver Manager** (see “Choosing the Values of Variables Not Solved For” on page 438 and “The Linear or Linearized Model” on page 386).

Normalization of the Eigensolutions

You can multiply a solution to an eigenvalue problem with an arbitrary nonzero number and still have an eigensolution. COMSOL Multiphysics normalizes the eigensolutions so that their discrete root-mean-square norm is 1, that is,

$$\frac{1}{N} \sum_{i=1}^N |U_i|^2 = 1$$

where U_i are the degrees of freedom solved for.

Ordering of the Eigensolutions

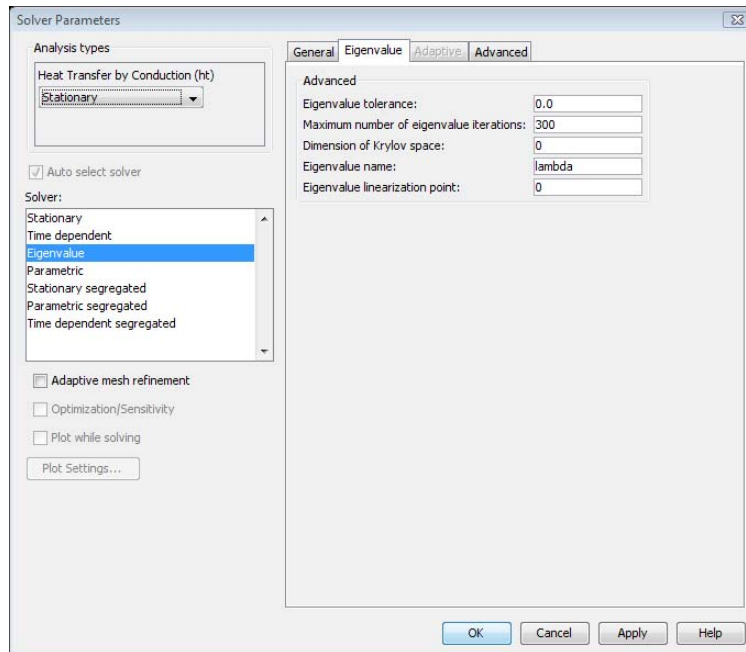
When you have solved an eigenvalue problem, the eigenvalues for the eigensolutions found appear sorted in the dialog boxes for postprocessing. COMSOL Multiphysics uses two sorting criteria:

- Increasing order of the real part of the eigenvalue, $\text{Re } \lambda$
- Increasing order of the absolute value of the imaginary part of the eigenvalue, $|\text{Im } \lambda|$

If the eigenvalue problem is linear—that is, if the matrix E is zero—the software uses the first sorting criterion; if that criterion does not separate eigenvalues, the second criterion is used. If the eigenvalue problem that you solve is quadratic—that is, if the matrix E is nonzero—the software uses the second sorting criterion; if that criterion does not separate eigenvalues, the first criterion is used.

EIGENVALUE SOLVER SETTINGS

To configure the advanced eigenvalue solver settings, click the **Eigenvalue** tab. Often you do not need to change these settings.



Settings for the eigenvalue solver.

The values in the **Eigenvalue tolerance** edit field controls the relative error in the computed eigenvalues. A zero value (default) means that the solver uses the machine's internal precision.

The eigenvalue solver is an iterative algorithm. The value in the **Maximum number of eigenvalue iterations** edit field limits the number of iterations (default = 300).

The **Dimension of Krylov space** edit field controls the algorithm’s memory use. The default value of 0 means that the solver sets the dimension of the Krylov space to approximately twice the number of eigenvalues.

By default, the name of the eigenvalue is `lambda` when it appears in equations and postprocessing expressions. You can change this name in the **Eigenvalue name** edit field. The change affects both the interpretation of equations and the name of the eigenvalue in postprocessing.

Note: All application modes that use the eigenvalue explicitly in their equations expect the name of the eigenvalue to be `lambda`. Changing the name in the **Solver Settings** dialog box does not update the application mode equations.

COMSOL Multiphysics reduces general nonlinear eigenvalue problems to a quadratic approximation around the value specified in the **Eigenvalue linearization point** field. The shift specified in the **Search for eigenvalues around** field is compensated accordingly. Therefore, changing the linearization point has no effect at all for linear or quadratic eigenvalue problems.

Eigenfrequencies vs. Eigenvalues

For many applications the eigenfrequency, f —or the corresponding angular frequency, ω —is a more interesting quantity than the eigenvalue, λ . Typically, these quantities are related as

$$\omega = 2\pi f = i\lambda$$

In some application modes you can select whether to work with eigenvalues or eigenfrequencies, and sometimes also angular frequencies.

The Eigenvalue Solver Algorithm

For a description of the algorithm used by the eigenvalue solver, see “The Eigenvalue Solver Algorithm” on page 536 in the *COMSOL Multiphysics Reference Guide*.

The Parametric Solver

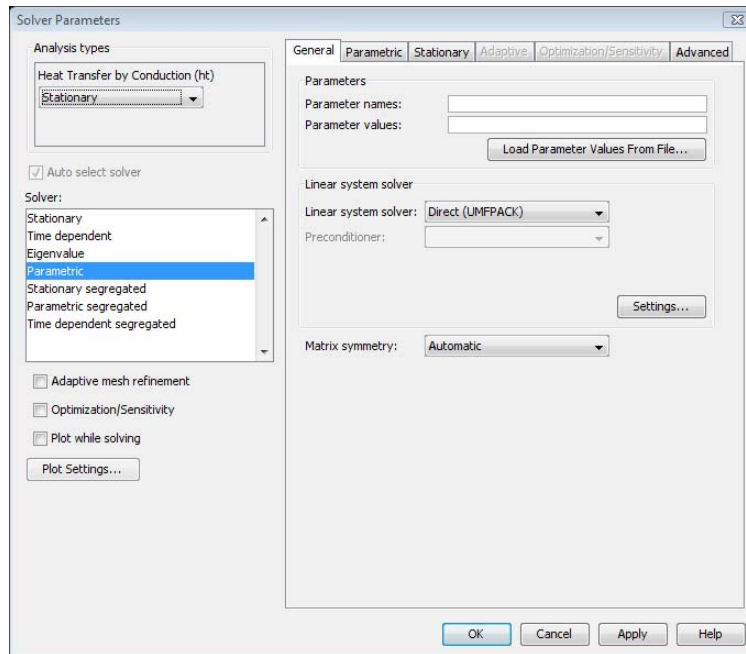
Use the parametric solver to find the solution to a sequence of linear or nonlinear stationary PDE problems that arise when you vary some parameters of interest. The parametric solver can also prove useful when it is difficult to get convergence in a nonlinear model. You can then introduce a parameter 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, giving the nonlinear solver a good initial guess based on the solution for the previous parameter value. The parametric solver is sometimes called a *continuation method* or a *path-following method*. See the section “Selecting a Stationary, a Time-Dependent, or an Eigenvalue Solver” on page 380 to decide if the parametric solver should work for a given model.

For parametric sweeps that perform several runs using the currently selected solver while varying a set of model parameters, see “Setting Up Parametric Sweeps” on page 383

USING THE PARAMETRIC SOLVERS

To use the parametric solver, go to the **Solver Parameters** dialog box and select **Parametric** from the **Solver** list.



Selecting the parametric solver.

Specifying the Parameters and Their Values

Specify the parameters to vary in the **Parameter names** edit field. The parameters can occur in equations, constraints, or expressions. If you are running COMSOL

Multiphysics in distributed mode (see “Distributed-Memory Parallel COMSOL” on page 74 of the *COMSOL Installation and Operations Guide*), the software distributes the parameter sweep on the nodes; it then assumes that the parameters are independent in the sense that the program can split the list in any way without causing convergence problems for each separate problem.

Note: You cannot vary parameters that control the geometry, mesh, or initial values through the parametric solver. To vary a geometric parameter, use the Parameterized Geometry application mode (in 2D) or use scripting to vary the geometry and compute the solution in a loop. If you have licenses for the CAD Import Module and SolidWorks or Autodesk Inventor, you can vary the geometry automatically by using the SolidWorks or Autodesk Inventor bidirectional interface. For further information about this feature, see “Bidirectional Interface to CAD Software” on page 30 of the *CAD Import Module User’s Guide*.

The list you enter in the **Parameter values** edit field is a vector. For a single parameter, it must contain a monotonically increasing or decreasing sequence of real parameter values, for example, `0:0.1:1.5` or `linspace(1,100,25)` (see “Entering Vector-Valued Expressions” on page 145 for more information on how to enter a vector of values).

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** edit 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** edit field. Then use the entry `1 7 8 2 4 12 3 1 6` in the **Parameter values** edit 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.

You can also use the **Load Parameter Values From File** option. This option loads the values from a text file into the **Parameter values** edit field. In this text file, the parameter sets must be listed row-wise with the values in each set separated by a space or tab.

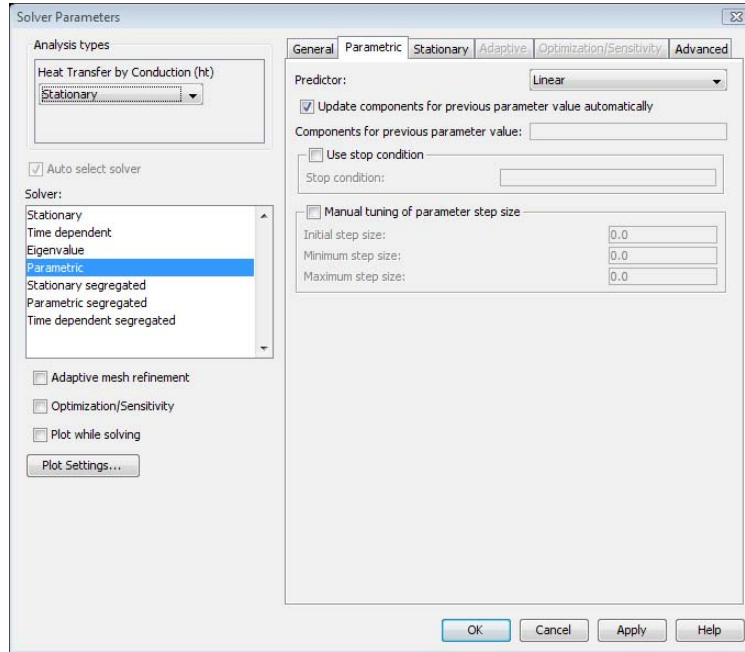
If you have already assigned some of the parameters values in the **Constants** dialog box, the values in the **Parameter values** edit field take precedence when using the parametric solver. If you supply more than two parameter values or parameter-value sets, the solver computes solutions for all of them. In the single-parameter case, the algorithm also solves for intermediate parameter values if this is needed to reach convergence, but the intermediate solutions do not appear in the output. If you supply exactly two values, the output also contains the solutions for all intermediate parameter values that the solver uses.

Setting Other Solver Properties

- The selection of analysis type sets up a default for the linear system solver. The section “Selecting a Linear System Solver” on page 426 describes how to override the default settings.
- For symmetric matrices, use the criteria from the section “Which Problems Are Symmetric?” on page 431.
- When using the parametric solver for a nonlinear model, you can tune the nonlinear solver settings on the **Stationary** page (see “Nonlinear Solver Settings” on page 389).
- If the parametric analysis contains a large number of parameter steps in the results data, click the **Advanced** tab and select the **Store solution on file** check box. For more information about this option, see “Storing Solutions on File” on page 526 of the *COMSOL Multiphysics Reference Guide*.

THE PARAMETRIC SOLVER SETTINGS

To specify additional settings for the parametric solver, click the **Parametric** tab. Usually you need not change these settings.



The parametric solver settings.

Selecting a Prediction Method

If you select **Linear** in the **Predictor** list (the default) and you are solving for one parameter, COMSOL Multiphysics obtains an initial solution guess at the new parameter value by following the tangent to the solution curve at the old parameter value when the solver steps from one value to another. If you select **Constant** or you are solving for several parameters, COMSOL Multiphysics instead takes the initial guess as the solution for the old parameter value. Linear prediction is usually the most efficient method for one parameter.

Component for Previous Parameter Value

For some applications, such as contact problems with friction in structural mechanics, you need access to the solution at a previous parameter value. The solution components you enter in the **Components for previous parameter value** edit field are treated in a separate linear solution step (or updating step) that the solver performs

after it has finished solving for the current parameter value. These components are subtracted from the solution components and are not included in the main solver step. Let the main components denote the solution components excluding the components from previous step, and let p_i denote the parameter value in the parameter step i . Then the following algorithm describes the procedure:

- 1 Set $i = 1$. Initialize all the solution components.
- 2 Solve for the main components and the parameter value, p_i , while keeping the components from previous step fixed.
- 3 Make a linear solve for the components from the previous step, while keeping the main components fixed.
- 4 If p_i is the last parameter, the procedure is complete, else let $i = i + 1$ and go to Step 2.

The linear system solver used in the linear solver (Step 3) is UMFPACK.

If you select the **Update components for previous parameter value automatically** check box, the application modes fill in these components automatically. For applications in contact modeling, see the *Structural Mechanics Module User's Guide*.

Note: Entering variables in the **Components for previous parameter value** edit field does not formulate an updating equation. You must formulate the updating equations in the standard way through the application modes.

Specifying a Stop Condition

You can force the solver to stop based on a stop condition. To do so, select the **Use stop condition** check box. The solver then stops before the expression in the **Stop condition** edit field becomes negative. For more information, see “Manual Tuning of the Nonlinear Solver” on page 396 in the section about the time-dependent solver.

Manual Tuning of the Parameter Step Size

By selecting the **Manual tuning of parameter step size** check box, you control how many intermediate parameter values the algorithm solves for:

- The value in the **Initial step size** edit field is a positive number that determines the magnitude of the first parameter step. The default value of 0 means that the initial step equals the difference between the two first values in the **Parameter values** edit field. If the initial time step is too big, the solver might not converge after the first step, and it gradually lowers the step size until it obtains convergence. On the other

hand, if the initial step size is too small, the solver might take an unnecessarily large number of small parameter steps.

- The value in the **Minimum step size** edit field is a safeguard against parameter steps that are too small. The default value of 0 means that the minimum step size is 1/200 of the parameter span. If the nonlinear Newton iteration diverges for a certain parameter value, the software reduces the step size, but if it goes below the minimum step size, the solver terminates with a warning. In this case the solution process might have reached some kind of singularity (a turning point or bifurcation) beyond which the continuation of the solution does not exist or is non-unique.
- The value in the **Maximum step size** edit field is an upper bound on the parameter step size. The default value of 0 means that the maximum step size equals the parameter span. Decrease this value if you suspect that the solver takes unnecessarily long steps and thereby switches branches or misses a singularity (alternatively, use more parameter values in the **Parameter values** edit field).

The Parametric Solver Algorithm

See the section “The Parametric Solver Algorithm” on page 537 of the *COMSOL Multiphysics Reference Guide*.

The Stationary Segregated Solver

Use the stationary segregated solver for linear or nonlinear stationary PDE problems when you want to split the solution steps into substeps. You define the different substeps by grouping solution components’ names together. In a solution step for a substep, the segregated solver uses the damped Newton method and computes only the Jacobian related to the solution components, a procedure that can save both memory and assembly time. Also, it is possible to choose which linear system solver to use in a substep independently from other substeps. For problems where a full Newton approach does not converge, a segregated solution approach can sometimes work well. On the other hand, in the vicinity of a solution where the Newton approach converges quadratically, the segregated solver approach often converges more slowly.

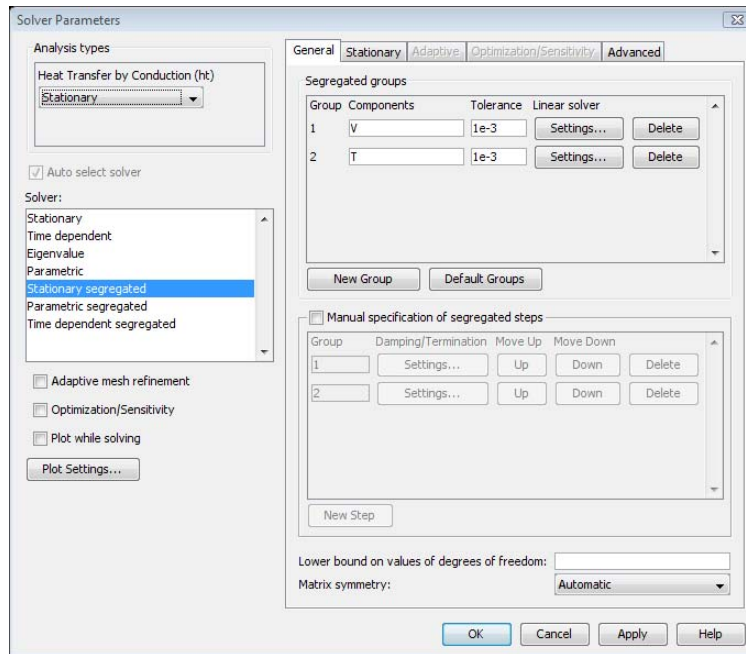
For an example of how to use the stationary segregated solver, see “Example—Resistive Heating” on page 395 of the *COMSOL Multiphysics Modeling Guide*. Thermal stress models in the Structural Mechanics Module and fluid-flow models in the Chemical Engineering Module are other examples of models that benefit from using the segregated solver.

The Thermal-Structural Interaction and Thermal-Electric-Structural predefined multiphysics couplings in the Structural Mechanics Module and the MEMS Module

use predefined settings for the stationary segregated solver that efficiently solve these applications where there is a one-way coupling from the thermal analysis to the structural analysis.

USING THE STATIONARY SEGREGATED SOLVER

To use the stationary segregated solver, go to the **Solver Parameters** dialog box and select **Stationary segregated** from the **Solver** list.



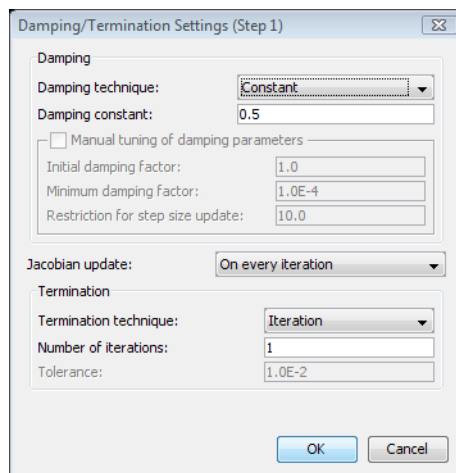
The General page settings for the stationary segregated solver in the Solver Parameters dialog box.

To generate the segregated solver groups, click the **New Group** button and provide the component names in the **Components** edit field. COMSOL Multiphysics predefines segregated solver groups, one for each application mode. The component names in each solver group are the dependent variables in the application mode corresponding to that group. You can edit, add, and rearrange the contents of these predefined groups. To remove a segregated solver group, click the **Delete** button. For each group you can choose the linear system solver independently of the other groups. To select or adjust linear system solver, click the corresponding **Settings** button (see also the section “Selecting a Linear System Solver” on page 426).

To control the order in which the segregated solver solves for the segregated group components, in its *segregated substeps*, select the **Manual specification of segregated steps** check box. Each line corresponds to a substep in the segregated algorithm. One *segregated iteration* is defined as performing all the substeps. In the **Group** edit field you provide the segregated solver group number for the corresponding substep. To create additional substeps, click the **New Step** button, and to remove a substep, click the **Delete** button for that substep. Use the **Up** and **Down** buttons to move the corresponding substep up or down in the list of substeps.

DAMPING, JACOBIAN UPDATE, AND TERMINATION FOR THE SUBSTEPS

In each substep, the segregated solver uses a damped Newton method. By default, one iteration using an updated Jacobian and with damping factor 0.5 is used for each substep. If the **Manual specification of segregated steps** check box is selected, you can specify the damping and termination techniques and how often the Jacobian is updated for each substep independently from the other substeps. Clicking the **Settings** button for a substep opens the **Damping/Termination Settings** dialog box for that substep.



The Damping/Termination Settings dialog box for the first substep.

In the **Damping technique** list, you can choose between the techniques **Constant** and **Automatic**. The choice **Constant** results in the damped Newton method with a constant damping factor that you enter in the **Damping constant** edit field. With the choice **Automatic**, a damped Newton method with automatic choice of damping factor is used. This is the same damped Newton method that is the default method in the coupled

solver. You can control the automatic choice of damping factor to some extent when the **Manual tuning of damping parameters** check box is selected (see “Manual Tuning of Damping” on page 390).

In substeps where you have chosen **Constant** as **Damping technique**, you can also control how often the Jacobian is updated by choosing between **Minimal**, **On first subiteration** and **On every iteration** in the **Jacobian update** list. With **Minimal**, the Jacobian for the substep is computed only once; this Jacobian is then reused in all subsequent iterations for the substep. With the choice **On first subiteration**, the Jacobian is updated once every segregated iteration. By choosing **On every iteration**, the Jacobian of the substep is updated on every Newton iteration. In substeps with **Automatic** as **Damping technique**, you cannot control the frequency of the Jacobian update; the automatic choice of damping factor requires the Jacobian to be updated on every Newton iteration.

You choose the termination criteria for the Newton iterations for each substep from the **Termination technique** list. If you choose **Iteration**, the fixed number of Newton iterations in the **Number of iterations** edit field are performed. With the choice **Tolerance**, the Newton iterations are continued until the estimated relative error for the group is smaller than the number entered in the **Tolerance** edit field. Also, if the relative error is not below the tolerance after the maximum number of allowed iterations, specified in the **Maximum number of iterations** edit field, the segregated solver terminates with an error message. A combination of the two termination techniques described above is obtained if you choose **Iterations or tolerance**; the Newton iterations are continued until either the estimated error is smaller than the number entered in the **Tolerance** edit field or until the number of iterations specified in the **Number of iterations** edit field have been performed. Note that termination of the Newton iterations without reaching the specified tolerance does not result in termination of the segregated solver when **Iterations or tolerance** is used.

TERMINATION TECHNIQUES FOR THE SEGREGATED SOLVER

You choose the criteria for termination of the segregated solver from the **Termination technique** list on the **Stationary** page. As for the termination of the Newton iterations for each substep, you can choose between **Iterations**, **Tolerance**, and **Iterations or tolerance**. With the choice **Iterations**, the fixed number of segregated iterations entered in the **Number of iterations** edit field (on the **Stationary** page) are performed. Using a single iteration is an efficient method for one-way coupled multiphysics problems but does not provide a converged solution for multiphysics problems where the couplings appear in both directions between the segregated groups. When **Tolerance** is chosen,

the segregated solver terminates if a convergence criterion is fulfilled. The value in the **Tolerance** edit field (default = 10^{-3}) for each group on the **General** page gives the convergence criterion. The segregated-solver iterations stop when, for all groups, the relative error estimate is smaller than the corresponding tolerance. For details on the error estimate, see the “The Stationary Segregated Solver Algorithm” on page 537 in the *COMSOL Multiphysics Reference Guide*. Also, if the number of segregated iterations exceeds the number in the **Maximum number of segregated iterations** edit field (on the **Stationary** page), the segregated solver terminates with an error message. The termination technique **Iterations or tolerance** terminates the segregated iterations if either the convergence criteria of the termination technique **Tolerance** is met or if the number of segregated iterations exceeds the number in the **Number of iterations** edit field.

LOWER BOUND ON THE DEGREES OF FREEDOM

For the segregated solver there is a possibility to directly—that is, without any regards to the equations—impose a restriction on the degrees of freedoms. To use this feature, edit the value in the **Lower bound on values of degrees of freedom** edit field and enter a space-separated list whose entries alternate between component names (the names of the degrees of freedom) and limiting values, for example, `c1 0 c2 1e-3`. For these pairs, the solution vector is modified after each substep in such a way that

$$U_j = \max(U_j, U^L)$$

where U_j are the corresponding degrees of freedoms and U^L denotes the limiting value. Note that if either U_j or U^L is complex valued the real part of that quantity is used in the above equation.

The selection in the **Matrix symmetry** list applies to all the segregated solver groups. For the **Automatic** choice, the solver can detect and make use of symmetry for the group Jacobian independently of other groups.

Note: The segregated group Jacobians can be symmetric or Hermitian even if the fully-coupled problem results in a nonsymmetric or non-Hermitian Jacobian.

The selection in the **Linearity** list on the **Stationary** page applies to all the segregated solver groups. This selection is not as important as it is for the standard stationary solver because the stationary segregated solver uses the same iterative procedure both for linear and nonlinear problems and always checks the error criteria. The associated

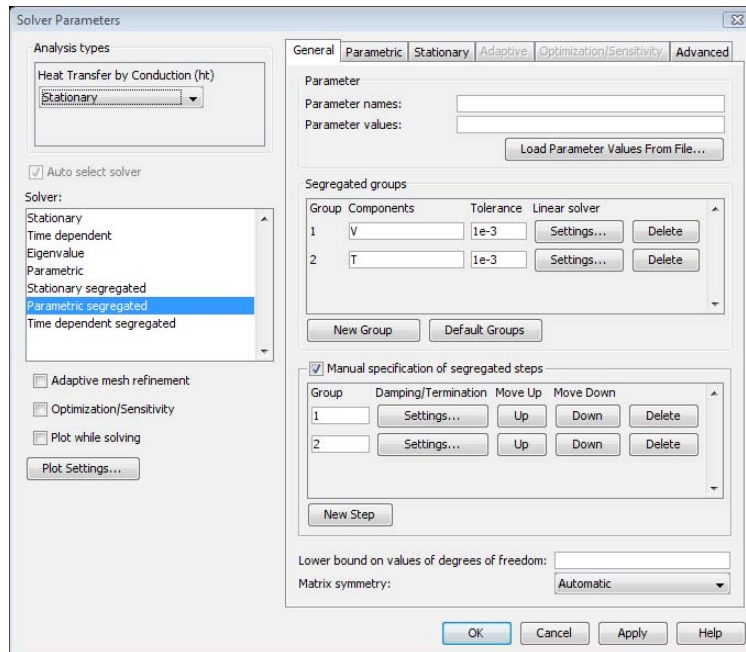
settings do affect which group Jacobians that are reassembled, and an incorrect selection can therefore result in suboptimal convergence.

The Parametric Segregated Solver

Use the parametric segregated solver to find the solution to a sequence of linear or nonlinear stationary PDE problems that arise when you vary a parameter of interest. The motivation to use a segregated solver approach instead of the standard parametric solver is similar to the motivation to use the stationary segregated solver instead of the standard stationary solver; see “The Stationary Segregated Solver” on page 411.

USING THE PARAMETRIC SEGREGATED SOLVER

To use the parametric segregated solver, go to the **Solver Parameters** dialog box, find the **Solver** list, and select **Parametric segregated**.



Selecting the parametric segregated solver.

For details about how to specify the parameter and its values as well as the settings on the **Parametric Solver Settings** page, see “The Parametric Solver” on page 405. There

are some minor differences between the parametric solver and the parametric segregated solver in the way COMSOL Multiphysics executes the parameter stepping:

- The parametric segregated solver can only fail for a parameter value when the segregated iteration number exceeds the value in the **Maximum number of segregated iterations** edit field, whereas the standard parametric solver can also fail when the damping factor is too small. This can make the standard parametric solver more efficient for certain problems where there are several solver failures.
- After a successful step, the parametric solver chooses the next parameter step size based on the convergence speed of the Newton iteration. The parametric segregated solver uses a constant factor of 2 to increase the parameter step size.

The Time-Dependent Segregated Solver

In the same way as it is more efficient to solve some stationary PDE problems if the solution components are split into different groups (see “The Stationary Segregated Solver” on page 411), a segregated approach for the linear or nonlinear system of equations that is solved in each time step can sometimes be more efficient. For such problems, use the time-dependent segregated solver.

USING THE TIME-DEPENDENT SEGREGATED SOLVER

To use the time-dependent segregated solver, go to the **Solver Parameters** dialog box, find the **Solver** list, and select **Time dependent segregated**.

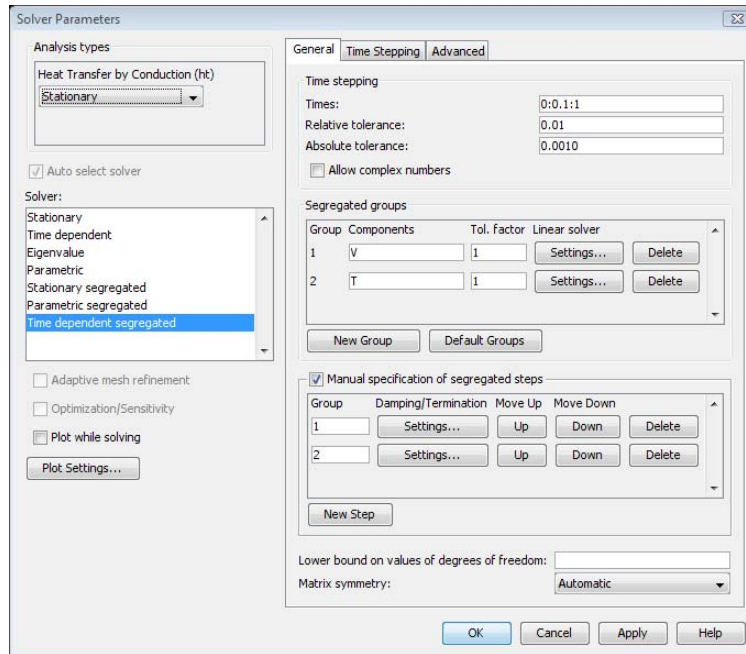


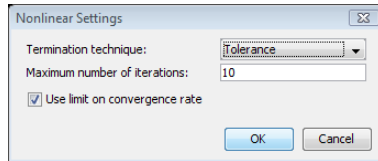
Figure 6-1: Settings for the time-dependent segregated solver.

If you are familiar with the time-dependent solver and the stationary segregated solver, you will be familiar with most settings in the time-dependent segregated solver. How to set the time step and tolerances is explained in “The Time-Dependent Solver” on page 391. That section also contains information about the settings on the **Advanced** page and most settings on the **Time Stepping** page. For information on how to specify segregated groups and steps, as well as the different termination techniques of the segregated solver, see “The Stationary Segregated Solver” on page 411. There are, however, some minor differences:

- With **Manual specification of segregated steps** selected, clicking the **Settings** button for a step opens the **Damping/Termination Settings** dialog box, where the **Jacobian update** list is located. For the time-dependent segregated solver, there is an additional choice in this list compared to the stationary segregated solver: **Once per time step**,

in which case the Jacobian matrix of the step is updated on the first subiteration of each time step.

- On the **Time Stepping** page, the dialog box that opens when you click the **Nonlinear Settings** button is different from the corresponding dialog box for the time-dependent solver. This is because **Damping technique** and **Jacobian update** are specified per step in the time-dependent segregated solver (in the **Damping/Termination Settings** dialog box). For the time-dependent segregated solver, only the termination of the segregated solver is set in the **Nonlinear Settings** dialog box, in the same way as for the stationary segregated solver; see “Termination Techniques for the Segregated Solver” on page 414.



Nonlinear Settings dialog box for the time-dependent segregated solver.

Adaptive Mesh Refinement

When modeling, it is important to select a mesh that minimizes the errors in the quantities of interest. COMSOL Multiphysics provides several ways to set the mesh size manually (see Chapter 5, “Meshing,” on page 299). However, it is not easy to minimize the error in a desired quantity by manually specifying a mesh. In many applications, the algorithm must resolve the solution in great detail only on small portions of the domain (such as boundary layers, shocks, small geometrical details, or localized loads). For such situations, consider using adaptive mesh generation, which identifies the regions that require a high resolution and produces an appropriate mesh.

The adaptive solver works for linear and nonlinear *stationary PDE problems* (including parameter-dependent problems) and for *eigenvalue PDE problems*. It does not support quadrilateral, hexahedral (brick), or prism meshes. For models that include several geometries, COMSOL Multiphysics applies adaptive mesh generation to only one of them.

For a description of the algorithm on which adaptive mesh refinement in COMSOL Multiphysics is based, see the section “The Adaptive Solver Algorithm” on page 538 of the *COMSOL Multiphysics Reference Guide*.

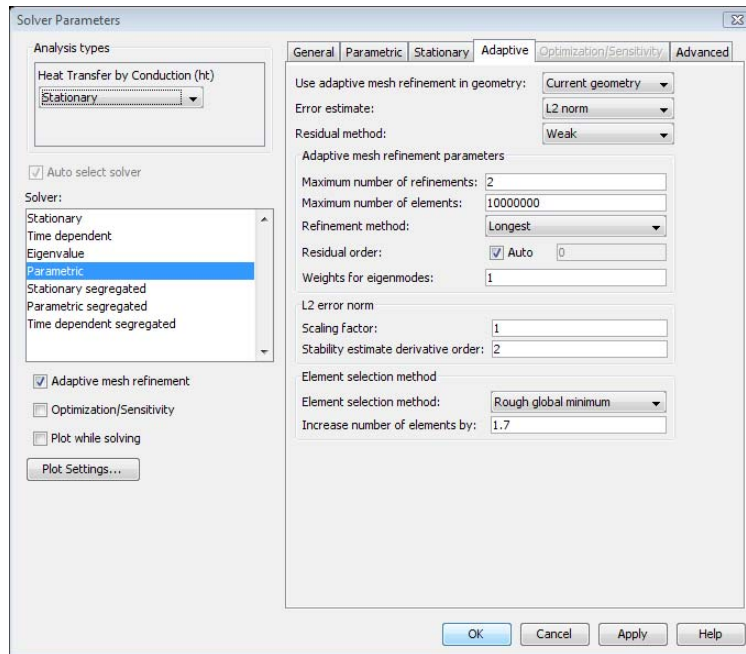
USING ADAPTIVE MESH REFINEMENT

To use adaptive mesh refinement, open the **Solver Parameters** dialog box and select the **Adaptive mesh refinement** check box. Adaptive mesh refinement works with all solvers except the time-dependent solver. After clicking **Solve**, the software returns with the adapted mesh and the corresponding solution (the original mesh disappears).

You can adjust the settings for adaptive mesh refinement on the **Adaptive** page.

Note: The settings on the **Adaptive** page are with reference to the current mesh when you start the solver, if one exists. If no mesh exists when the solver starts, the software generates an initial mesh using the settings in the **Free Mesh Parameters** dialog box.

If your model contains more than one geometry, you can select the geometry for the adaptive solver from the **Use adaptive mesh refinement in geometry** list at the top of the page. The default setting is **Current geometry**.



Settings for adaptive mesh refinement.

For the stationary and parametric solvers you can, in the **Error estimate** list, select which norm to use for the error estimate. The **L2 norm** is the default and normally needs no further tuning or settings. The **Functional**-based norm uses a user-specified functional, which you supply in the **Functional** edit field. You can, for example, give the name of an integration coupling variable that integrates the solution over a subdomain or a boundary. You can also give an expression involving several variables. The only limitation is that the expression must have global scope. The functional-based norm adapts the mesh toward improved accuracy in the expression for the functional, so choose it with some care. For the eigenvalue solver, only the **L2 norm** is available.

In the **Residual method** list you can select the method to use when computing the PDE residual. This setting is included for possible backward compatibility issues, so for new models you can safely use the default method, **Weak**, which supports all solution forms. (The method **Coefficient** only supports the **General** and **Coefficient** solution forms; see the section “Solution Form” on page 526 of the *COMSOL Multiphysics Reference Guide*.)

The **Auto** check box for **Residual order** is selected per default. The software then automatically determines the order of decrease in equation residuals on the basis of the shape function orders in the model. Roughly speaking, the order is one less than the shape function order for the corresponding equation. If you clear this check box, you can provide this order in the edit field as a nonnegative integer.

Terminating Adaptive Mesh Generation

The adaptive solver terminates when it has obtained solutions for the number of refined meshes you specify in the **Maximum number of refinements** edit field. In 2D the default value is 2, which means that the software solves on the original mesh along with two refined meshes. In 3D the default value is 1, and in 1D it is 5. If the number of mesh elements exceeds the value specified in the **Maximum number of elements** edit field (default = 10,000,000), the solver terminates even if it has not reached the specified maximum number of refinements.

Adaptive Mesh Refinement Methods

The refinement method determines how the algorithm refines the elements it selects on the basis of the settings in the **Element selection method** area. If you choose **Longest** from the **Refinement method** list (not available for 1D geometries), the adaptive solver refines only the longest edge of each selected element; if, instead, you select the option **Regular** (not available for 3D geometries), the solver refines these elements in a regular pattern. For both methods, some additional adjacent elements are refined to get a conformal mesh, that is, a mesh without hanging nodes. The default method in 2D is

Longest. A third option, **Meshinit**, generates a completely new mesh where the mesh-element size at a particular location is roughly equal to that of the old mesh divided by a refinement factor determined by the settings in the **Element selection method** area.

Error Estimation Settings

For eigenvalue problems, the error estimate is a weighted sum of the error estimates for the various eigenmodes. You can enter a space-separated list of positive weights in the **Weights for eigenmodes** edit field (only the relative weights matter). The default value of 1 means that all the weight is put on the first eigenmode.

L2 norm settings The **Scaling factor** edit field takes a space-separated list of scaling factors, one for each PDE in the system. When computing the local error estimate, the algorithm divides the error estimate for each PDE by the scaling factor before summing them. The default value of 1 means that all PDEs get the same weight in the error estimate. The L2 norm error estimate is based on a stability estimate for the PDE. You can specify its order in the **Stability estimate derivative order** edit field (default = 2).

Functional settings The **Functional** edit field takes the name of the variable used for the error estimate. For this method the mesh refinement is oriented toward reducing the error in this variable. For this method to work, this variable must be a global variable and depend on the solution in the geometry used for the adaptive solver.

Element Selection Settings

COMSOL Multiphysics provides three ways to select which elements to refine; choose one of the following in the **Element selection method** list:

- **Rough global minimum** (the default) minimizes the error's L2 norm 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 **Increase number of elements by** edit field. The default factor is 1.7, that is, a 70% increase. With this option, the software can refine some elements with a factor larger than 2.
- **A fraction of worst error** isolates those elements that have a local error indicator larger than a given fraction of the largest local error indicator. Enter this fraction in the **Worst element fraction** edit field (default = 0.5). The software refines the indicated elements just once.
- **A fraction of elements** refines a given fraction of the elements. Enter this fraction in the **Element fraction** edit field (default = 0.5). The software refines the selected elements just once.

By including a Sensitivity Analysis application mode in a model, you can analyze the sensitivity of a stationary PDE problem's solution variables or a scalar objective function with respect to a set of sensitivity variables that you specify; Chapter 10, "Sensitivity Analysis," of the *COMSOL Multiphysics Modeling Guide* gives some background on sensitivity analysis and describes this application mode in detail. The current section focuses on the solver settings you can make when performing a sensitivity analysis.

If you have the Optimization Lab (an optional add-on), you can also set up and solve optimization problems in the COMSOL Multiphysics user interface using the Optimization application mode. Chapter 11, "Optimization," of the *COMSOL Multiphysics Modeling Guide* describes this application mode, including solver-related settings, and it will not be discussed further in this section.

SENSITIVITY ANALYSIS

You can perform sensitivity analyses in combination with all stationary and parametric standard solvers—often referred to as *forward* solvers in this context. Specify the settings for the relevant forward solver according to the guidelines given elsewhere in this chapter.

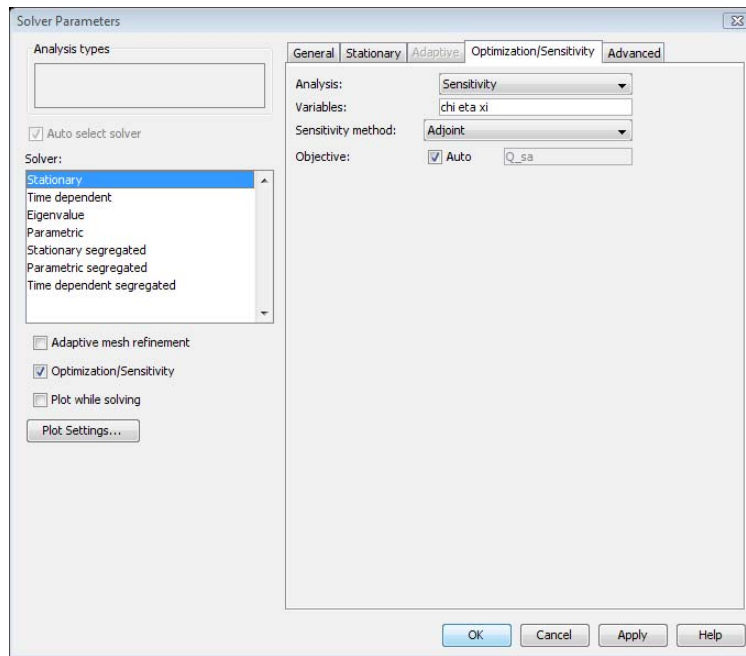
When you have defined a set of sensitivity variables and (optionally) an objective function in the dialog boxes for the Sensitivity Analysis application mode and have chosen a forward solver and specified any associated settings for your PDE problem, you are ready to set up the sensitivity analysis.

Begin by selecting the **Optimization/Sensitivity** check box below the **Solver** list in the **Solver Parameters** dialog box to activate the page with the same name. On the **Optimization/Sensitivity** page, use the default setting in the **Analysis** list, which is **Sensitivity** (the only analysis type available if you do not have the Optimization Lab). In the **Variables** edit field, all sensitivity variables you have defined appear by default; you can modify the entry in this edit field to run an analysis for any subset of the sensitivity variables. Any sensitivity variables not included in the **Variables** edit field retain their initial values during the sensitivity analysis.

Next, select one of the options in the **Sensitivity method** list: **Adjoint** (default) or **Forward**. Choose the adjoint sensitivity method to compute the sensitivity of the objective function only. To evaluate the sensitivities of all solution variables for the PDE problem and an objective function (which is then optional), you need to choose the forward sensitivity method instead. For detailed descriptions of the two sensitivity

methods, see the section “The Sensitivity Solver” on page 295 of the *COMSOL Multiphysics Modeling Guide*. The underlying algorithms are described in the section “The Sensitivity Solver Algorithm” on page 541 of the *COMSOL Multiphysics Reference Guide*.

If you defined an objective function in the Sensitivity Analysis application mode, the sensitivity solver uses this function as the default objective. Alternatively, you can use any real-valued scalar expression in the model as the objective function. To enter an objective function, clear the **Auto** check box for the **Objective** and then type the desired expression in the associated edit field.



Settings for sensitivity analysis.

Plotting the Solution While Solving

You can visualize intermediate solutions as the solver completes them by selecting the **Plot while solving** check box. The plots use the current settings in the **Plot Parameters** dialog box (see “Visualization Using the Plot Parameters Dialog Box” in Chapter 7). Click the **Plot Settings** button to further specify when to plot, whether you want to

perform rougher plots, and if you want the figure window to close automatically once the solver finishes.

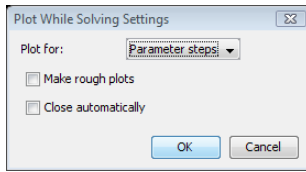


Figure 6-2: Plot settings for a nonlinear parametric problem.

The available plot levels depend on the solver (for example, stationary, parametric, or time dependent) and on the solver options, for example, whether optimization or adaptive mesh refinement is activated.

- **Mesh refinements** plots after each adaptive mesh refinement step.
- **Nonlinear steps** plots after each nonlinear (or segregated) iteration. If the stationary problem is linear, no plot is created.
- **Optimization steps** plots after each objective function evaluation of an optimization problem. When the optimization solver is selected, this is the only option.
- **Parameter steps** plots after each parameter step exported to the output.
- **Time steps** plots after certain time-steps, described further below.

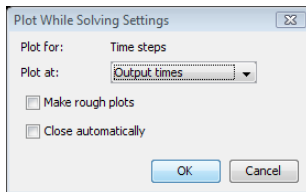


Figure 6-3: Plot settings for a time-dependent problem

- **Output times** plots at times that are exported to the output. (See “Setting the Time Steps for the Output and the Solver” on page 395.)
- **Time steps from solver** plots at all times taken by the solver.

Reference

1. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer-Verlag, 1991.

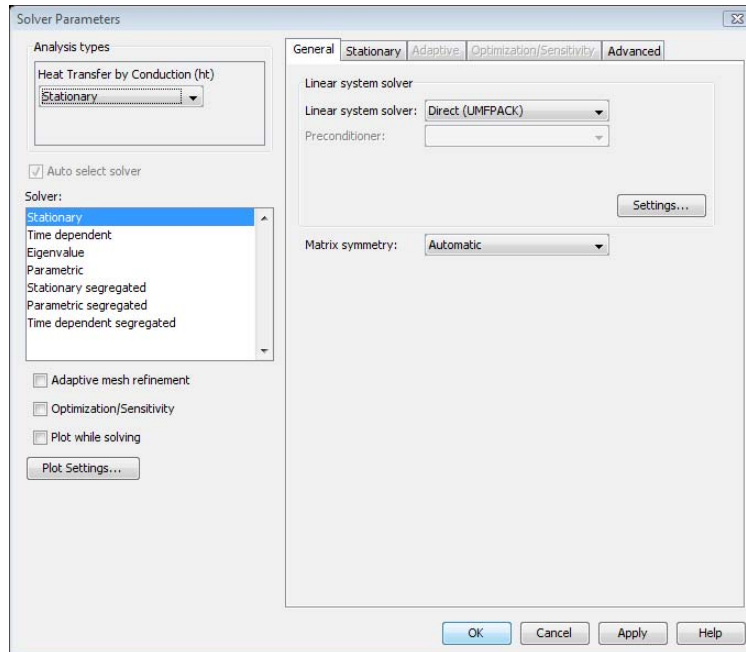
The Linear System Solvers

Whether working on a stationary, time-dependent, or eigenvalue model, the solver spends most of its time solving one or several systems of linear equations. Because the number of unknowns in these systems is usually very large, the most important solver parameter setting concerns the choice of linear solver.

This section provides some general guidelines to help you select the most effective linear system solver for your particular model. For detailed information about the solvers, preconditioners, and smoothers, as well as about the algorithms on which they are based, please consult the section “Linear System Solvers” on page 544 of the *COMSOL Multiphysics Reference Guide*.

Selecting a Linear System Solver

You control the linear system solver to use in the **Linear system solver** area on the **General** page of the **Solver Parameters** dialog box.



The Linear system solver area in the Solver Parameters dialog box.

In the **Linear system solver** list, you can choose between the following solvers, each of which is broadly categorized as direct or iterative:

LINEAR SYSTEM SOLVER	USAGE
DIRECT SOLVERS	
Direct (UMFPACK)	A highly efficient direct solver for nonsymmetric systems
Direct (SPOOLES)	An efficient direct solver for symmetric and nonsymmetric systems. It uses less memory than UMFPACK.
Direct (PARDISO)	A highly efficient direct solver for symmetric and nonsymmetric systems. It often uses less memory than UMFPACK.
Direct (PARDISO out of core)	An out-of-core version of PARDISO that stores the LU-factors on disk.
Direct Cholesky (TAUCS)	An efficient direct solver for symmetric, positive-definite systems
ITERATIVE SOLVERS	
GMRES	An iterative solver for nonsymmetric problems
FGMRES	An iterative solver for nonsymmetric problems. It can handle more general preconditioners but also uses more memory than GMRES.
Conjugate gradients	An iterative solver for symmetric positive definite problems
BiCGStab	An iterative solver for nonsymmetric problems. It uses a fixed amount of memory independent of the number of iterations. Therefore, it typically uses less memory than GMRES.
Geometric multigrid	An iterative solver for elliptic or parabolic problems

THE DIRECT SOLVERS

The *direct solvers* PARDISO, UMFPACK, SPOOLES, and TAUCS Cholesky are preferable for 1D and 2D models, and for 3D models with few degrees of freedom (roughly, fewer than 100,000 to 1,000,000 depending on available memory). You can access the TAUCS solvers only if the selection in the **Matrix symmetry** list is **Automatic** (the default) and the software detects the problem as symmetric, or if you select one of the options **Symmetric** or **Hermitian**. Use the out-of-core version of PARDISO for larger problems where the internal memory is insufficient.

Note: PARDISO is available on Linux, Windows, and Intel Mac. On Sun and PowerPC-based Macs, COMSOL Multiphysics uses SPOOLES instead.

The direct solvers solve a linear system by Gaussian elimination. This stable and reliable process is well suited for ill-conditioned systems. In 1D and 2D, where a direct solver is the default setting, these solvers require less tuning and are often faster than the iterative solvers. The elimination process sometimes requires large memory resources and long computing times, an effect particularly noticeable for 3D models.

THE ITERATIVE SOLVERS

For models with many degrees of freedom (roughly, more than 100,000 to 1,000,000 depending on available computer memory), the direct solvers typically need too much memory. In those cases, the more memory-efficient *iterative solvers* GMRES, FGMRES, Conjugate gradients, BiCGStab, and Geometric multigrid perform better. However, iterative solvers are less stable than direct solvers in that they do not always converge (arrive at a solution). To improve the convergence of an iterative solver (except Geometric multigrid), it is important to select an appropriate *preconditioner* from the **Preconditioner** list that appears if you select an iterative solver. The default setting for most 3D models calls on iterative solvers.

Linear System Solver Settings

Each linear solver has its own settings; to adjust them, click the **Settings** button to launch the **Linear System Solver Settings** dialog box. In this dialog box you can choose from a larger set of linear system solvers and preconditioners. However, the most appropriate solvers and preconditioners are all available in the **Linear system solver** area on the **General** page of the **Solver Parameters** dialog box. For the multigrid solvers/preconditioners it is also possible to select the smoothers and the coarse solver.

Linear System Solver Selection Guidelines

The ruling application mode selects a default linear system solver that usually is appropriate for the problem type, at least for single-physics models. If the default solver does not perform well, review the following guidelines to choose linear system solver.

- 1 Try the PARDISO direct solver if it is available on your system.
- 2 Try the UMFPACK direct solver.

- 3 If the UMFPACK solver runs out of memory or slows down due to swapping during the matrix-factorization phase, try the SPOOLES direct solver. It often uses less memory but is less numerically stable. SPOOLES is also slower.
- 4 If the system is real symmetric, complex symmetric, or complex Hermitian (see “Which Problems Are Symmetric?” on page 431), try a solver that takes advantage of that, such as the PARDISO or SPOOLES direct solver. They then save half the memory because they only store the upper triangular part of the matrix. If, in addition, the system is positive definite (see “Which Models Are Positive Definite?” on page 433), you can try the TAUCS Cholesky direct solver.
- 5 If the solver still runs out of memory or is too slow, use one of the iterative solvers GMRES, FGMRES, or BiCGStab. Select a preconditioner according to the guidelines in the following section.
- 6 If the system is positive definite and real symmetric or Hermitian, try the conjugate gradients iterative solver, which is more memory-efficient and sometimes faster than GMRES, FGMRES, and BiCGStab. Select a symmetric preconditioner as described in the guidelines in the following section.
- 7 Try the PARDISO out-of-core direct solver if it is available on your system and you run out of memory when using the other solvers or if the iterative solvers are not an option.

Selecting a Preconditioner

When using an iterative linear system solver you must select a preconditioner. This choice affects the number of iterations and the solver’s eventual convergence. Preconditioning can consume more time and memory than the actual iterative solver itself. To choose a preconditioner, click the **General** tab in the **Solver Parameters** dialog box and make a selection from the following entries in the **Preconditioner** list:

PRECONDITIONER	USAGE
Incomplete LU	For nonsymmetric systems
Geometric multigrid	For elliptic or parabolic systems
Incomplete Cholesky (TAUCS)	For symmetric positive definite systems
Algebraic multigrid	For scalar problems or loosely coupled multiphysics problems of the elliptic or parabolic type
SSOR	For large positive definite models
Diagonal scaling (Jacobi)	For large positive definite models

PRECONDITIONER	USAGE
SSOR vector	For large models using vector elements
Vanka	For large indefinite problems with zeros on the diagonal of the system matrix

For advice on how to determine to which category your model belongs, see “Guidelines for Classifying Your PDE Problem” on page 431. The concept of preconditioner is described in the section “The Preconditioned Linear System” on page 435.

Each preconditioner has its own settings; to adjust them, click the **Settings** button. For some preconditioners, the most important settings are also present in the **Linear system solver** area on the **General** page.

If, for some reason, you want to solve a model without a preconditioner, click the **Settings** button and select **None** from the **Preconditioner** list.

The Incomplete LU preconditioner works in a more general context than the others, but it might be impractical because of its time and memory requirements; when they work, the multigrid preconditioners are always preferable. The TAUCS incomplete Cholesky preconditioner is a more memory-efficient version of the Incomplete LU preconditioner for symmetric positive definite matrices. The SSOR and diagonal-scaling preconditioners use less time and memory but only ensure convergence of the iterative solver for positive definite problems. Problems with zeros on the diagonal are efficiently preconditioned with Vanka. To precondition electromagnetic problems that use vector elements for a PDE containing the curl-curl operator, use the SSOR vector preconditioner.

For detailed descriptions of the individual preconditioners, consult “Preconditioners for the Iterative Solvers” on page 553 of the *COMSOL Multiphysics Reference Guide*.

Preconditioner Selection Guidelines

The ruling application mode selects a default preconditioner that is usually appropriate for the problem type, at least for single-physics models. If the default does not perform well, select another one using the following guidelines:

- If the system is elliptic or parabolic (see “Elliptic and Parabolic Models” on page 434) use the Geometric multigrid preconditioner.
- If you solve an indefinite problem with zeros on the diagonal of the system matrix, such as the Navier-Stokes equations, try the Vanka preconditioner or the Geometric

multigrid preconditioner with Vanka or Incomplete LU as the smoother. With appropriate stabilization, it is possible in many cases to use SSOR as a GMG pre- and postsmoother instead of Vanka, which increases performance. This is the default setting in some fluid-flow application modes.

- If the system is positive definite but so large that the other preconditioners run out of memory, try the SSOR or Diagonal scaling (Jacobi) preconditioners.
- If you solve an electromagnetics problem using vector elements for a PDE containing the curl-curl operator, try the geometric multigrid preconditioner with the SOR vector pre-smoother and the SORU vector postsmoother, or try the SSOR vector preconditioner.
- Try the Incomplete LU preconditioner, which works for all linear systems. However, it requires the tuning of the drop tolerance; it can run out of memory; and in many cases it is not the most efficient preconditioner.
- If the system is elliptic or parabolic (see “Elliptic and Parabolic Models” on page 434) and the system is a real-valued PDE for a single solution component (that is, a scalar problem) you can alternatively try the Algebraic multigrid preconditioner.
- If the system is symmetric positive definite (see “Which Models Are Positive Definite?” on page 433) you can try the TAUCS incomplete Cholesky preconditioner. However, in this case Geometric multigrid likely performs better.

The Incomplete LU preconditioner and sometimes the multigrid preconditioners require some tuning to get fast convergence without running out of memory (see the appropriate sections on the following pages).

Guidelines for Classifying Your PDE Problem

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 discretizations 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 automatically 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 UMFPACK direct solver
- The Vanka preconditioner
- The Incomplete LU preconditioner
- The Algebraic multigrid solver/preconditioner

If you select the **Use Hermitian transpose of constraint matrix and in symmetry detection** check box on the **Advanced** page, the automatic symmetry detection checks if the matrices are Hermitian. If the matrices are symmetric or Hermitian, the solver prints this information in the Solver log, and if the linear solver can take advantage of symmetric or Hermitian matrices, the storage format for the matrices changes. If you use a solver that cannot take advantage of the symmetry, the Solver log contains a warning.

If you want to override the automatic symmetry detection, go to the **Solver Parameters** dialog box, click the **General** tab, and select **Symmetric**, **Hermitian**, or **Nonsymmetric** from the **Matrix symmetry** list.

Note: In general, models with multiphysics couplings create nonsymmetric matrices, and selecting **Symmetric** then leads to an incorrect solution. In some cases, you can exploit the symmetry in the single-physics parts by using the segregated solver or the Solver Manager to implement a sequential solution process, solving for one physics at the time.

Symmetry Criterion

If COMSOL Multiphysics solves a model using the general or coefficient solution form and the equations have no weak-form contributions, the following criterion determines whether the matrix is symmetric or not: Inspect the coefficients c , α , β , and a in the **Equation System>Subdomain Settings** dialog box and the coefficient q in the **Equation System>Boundary Settings** dialog box. The matrix is symmetric if

$$c_{lkji} = c_{klj}, \quad \alpha_{lkj} = \beta_{klj}, \quad a_{lk} = a_{kl}, \quad q_{lk} = q_{kl}.$$

The indices k and l refer to the rows and columns in the equation system, while the indices i and j refer to the space coordinates and their corresponding derivatives (see “The Linear or Linearized Model” on page 386). Further, the matrix is Hermitian if

$$\bar{c}_{lkji} = c_{kl ij}, \quad \bar{\alpha}_{lkj} = \beta_{klj}, \quad \bar{a}_{lk} = a_{kl}, \quad \bar{q}_{lk} = q_{kl}.$$

Symmetry Detection

COMSOL Multiphysics performs symmetry detection numerically. It detects a problem as symmetric if, for all matrices $A = \{A_{ij}\}$ involved (stiffness matrix, mass matrix, and so on), it holds that

$$|A_{ij} - A_{ji}| \leq \epsilon_s \sqrt{|A_{ij}|^2 + |A_{ii}| |A_{jj}|} \quad \text{for all } i \neq j$$

where ϵ_s is a symmetry tolerance. Similarly, the software detects a problem as Hermitian if, for all matrices $A = \{A_{ij}\}$ involved, it holds that

$$|A_{ij} - A_{ji}| \leq \epsilon_s \sqrt{|A_{ij}|^2 + |A_{ii}| |A_{jj}|} \quad \text{for all } i \neq j$$

and

$$|\text{Im}A_{ii}| \leq \epsilon_s |A_{ii}| \quad \text{for all } i$$

The symmetry tolerance ϵ_s is per default 10^{-10} .

WHICH MODELS ARE POSITIVE DEFINITE?

A model with a real symmetric or Hermitian system matrix is often also *positive definite*, which means that a number of efficient linear system solvers are applicable:

- The TAUCS Cholesky direct solver
- The Conjugate gradients iterative solver
- The TAUCS incomplete Cholesky preconditioner
- The PARDISO direct solver without row reordering and 2-by-2 Bunch-Kaufmann pivoting

Further, the simple preconditioners SSOR, SOR, SORU, diagonal scaling (Jacobi), and the multigrid solvers benefit from a positive definite matrix. A real symmetric or Hermitian matrix is positive definite if all its eigenvalues are positive.

For stationary problems, the system matrix is the Jacobian (stiffness) matrix A . This means that stationary models in diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix.

For time-dependent problems, the system matrix is of the form $A + \sigma B + \sigma^2 C$, where B is the damping matrix, C is the mass matrix, and $\sigma > 0$ is inversely proportional to the time step (if $C = 0$, then B is often called the mass matrix). Because these matrices are often positive definite, time-dependent models in diffusion, electromagnetics, structural mechanics, and heat transfer by conduction usually have a positive definite system matrix.

For eigenvalue problems, the system matrix is of the form $A - \sigma B + \sigma^2 C$, where σ is the *shift*, that is, the number around which the software searches for eigenvalues (specified in the **Search for eigenvalues around** edit field on the **General** page of the **Solver Parameters** dialog box; the default is 0). Because A , B , and C are usually positive definite, eigenvalue problems in acoustics, diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix if $\sigma \leq 0$.

If the model is stationary and you solve it with the general or coefficient solution forms, and if there are no weak-form contributions to the equations, the following sufficient criterion can determine whether the system matrix is positive definite: Inspect the coefficients c , α , β , and a in the **Equation System>Subdomain Settings** dialog box and the coefficient q in the **Equation System>Boundary Settings** dialog box. The system matrix is positive definite if $\alpha = \beta = 0$ and the c , a , and q matrices are positive definite or zero (however, not all of them should be zero).

ELLIPTIC AND PARABOLIC MODELS

The classes of elliptic and parabolic models include the positive definite models. For such models, the efficient multigrid preconditioners often perform well. A simplified definition of these classes reads as follows. A system of stationary or eigenvalue second-order PDEs is *elliptic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix. A system of time-dependent PDEs in general or coefficient form has a time derivative term of the form $d_a \dot{u}$, where the mass coefficient d_a is often a positive definite matrix and the e_a coefficient is 0. Such a system is called *parabolic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix.

Stationary or eigenvalue models in acoustics, convection-diffusion, electromagnetics, heat transfer, and structural mechanics are usually elliptic. Likewise, time-dependent models in convection-diffusion, electromagnetics, and heat transfer are often parabolic. In contrast, the Navier-Stokes equations, wave-type equations, or formulations involving weak constraints are neither elliptic nor parabolic.

If the software solves the model using the general or coefficient solution forms, and if the equations have no weak-form contributions, the following sufficient criterion can

determine whether a model is elliptic: If the coefficient c in **Equation System> Subdomain Settings** is a positive definite matrix, then the system is elliptic (if it is a stationary or eigenvalue model). In addition, if the model is time dependent and the mass coefficient d_a is a positive definite matrix, and $e_a = 0$, the system is parabolic.

The Preconditioned Linear System

If a linear system of equations is in the form $Ax = b$, the preconditioner is a matrix M that approximates A (sometimes in a very rough sense). Two possible ways of applying the preconditioner are

$$M^{-1}Ax = M^{-1}b \quad \textit{left preconditioning}$$

and

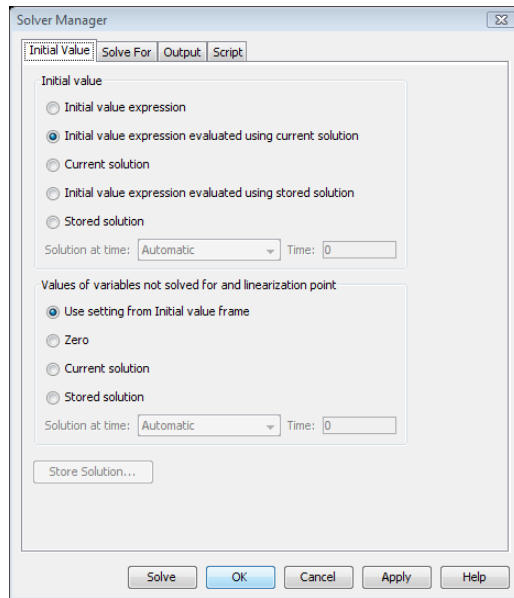
$$AM^{-1}u = b \quad \textit{right preconditioning},$$

where $u = Mx$. Because the system matrices $M^{-1}A$ and AM^{-1} are close to the identity matrix (in some sense), the iterative linear system solver converges faster when applied to the preconditioned linear system. For most preconditioners, the solver computes neither the matrix M nor the matrix M^{-1} ; instead, it provides a routine that computes the vector $M^{-1}y$ given the vector y . When using right preconditioning the variable u is never computed explicitly.

The Solver Manager and Solver Scripting

To help you control the solution process COMSOL Multiphysics provides the **Solver Manager** dialog box. In this dialog box, you specify the solution data to use as an initial value or linearization point, the solution components to solve for, and the solution components to output for postprocessing. You can also create a *script* of solver commands to automate a solution sequence (see “Solver Scripting” on page 443).

Initial Value Settings



Initial-value and linearization-point settings in the Solver Manager.

The **Initial Value** page controls how COMSOL Multiphysics computes the initial value for variables you instruct it to solve for, the values of variables not solved for, and the linearization point. Here you have access to both the *current solution* and the *stored solution*. By default COMSOL Multiphysics displays the current solution, which you

can save for future use by clicking the **Store Solution** button. Later, when once again computing an initial value or linearization point in the next solution process, COMSOL Multiphysics can access that stored solution. It is not possible, however, to access the stored solution for postprocessing purposes. For time-dependent, eigenvalue, and parametric analyses, you can choose to store only a subset of the solution in a multiple-selection list. The default settings store the entire solution.

Use the settings in the **Initial value** area to determine how COMSOL Multiphysics computes the value for the dependent variables (solution components) the software solves for at the beginning of the solution process (see “Choosing Variables for Which to Solve” on page 439). For a time-dependent problem this value determines the solution at the initial time; however, for a mixture of dynamic and stationary equations—a DAE system—the software modifies the initial value to become consistent with the equations (see “Settings for Differential-Algebraic Equations” on page 400). For stationary problems the initial value gives a first guess for the solution.

Settings in the **Values of variables not solved for and linearization point** area determine how COMSOL Multiphysics computes the values of the dependent variables (solution components) you choose not solve for. For the linear stationary solver and the eigenvalue solver, these settings also determine the *linearization point*, which is the value around which the software makes the linearization for the solution components it is solving for. If the model is linear, the linearization point is immaterial.

CHOOSING THE INITIAL VALUE

- Click the **Initial value expression** button to have the software calculate the initial value by evaluating the corresponding expressions on the **Init** pages of the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings** dialog boxes (all of which appear on the **Physics** menu). If these expressions depend on some solution component, the software uses a zero solution.
- Click the **Initial value expression evaluated using current solution** button (the default) to have the software calculate the initial value by evaluating the corresponding expressions on the **Init** pages of the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings** dialog boxes (all on the **Physics** menu). If these expressions depend on some solution component, the software uses the current solution for that solution component.
- Click the **Current solution** button to have the software take the initial value directly from the current solution.
- Click the **Initial value expression evaluated using stored solution** button to have the software calculate the initial value by evaluating the corresponding expressions on

the **Init** pages of the **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings** dialog boxes (all on the **Physics** menu). If these expressions depend on some solution component, the software uses the stored solution for that solution component.

- Click the **Stored solution** button to have the software take the initial value directly from the stored solution.

CHOOSING THE VALUES OF VARIABLES NOT SOLVED FOR

- Click the **Use setting from Initial value frame** button (the default) to have the software compute the values of solution components not solved for and the linearization point using the same settings as in the **Initial value** area.
- Click the **Zero** button to set the values of solution components not solved for and the linearization point to 0.
- Click the **Current solution** button to have the software take the value of solution components not solved for and the linearization point from the current solution.
- Click the **Stored solution** button to have the software take the value of solution components not solved for and the linearization point from the stored solution.

SPECIFYING THE CURRENT SOLUTION OR STORED SOLUTION

When working with a current solution or stored solution for a time-dependent, eigenvalue, or parametric analysis, you can choose which part of the solution to use.

For using these solutions in the **Initial value** area, the following options are available:

- Automatic selection (the default)
 - The last time step for solutions to time-dependent problems
 - The first eigenmode for eigenvalue problems
 - The last parameter value for parametric analyses
- Interpolation (for solutions to time-dependent problems only). Make an entry between the starting and ending times in the **Time** edit field.
- A solution in the output data. Choose one by selecting the corresponding time, eigenvalue, or parameter value in the list.

For using these solutions in the **Values of variables not solved for and linearization point** frame, the following options are available:

- The same options as for the initial value: automatic selection, interpolation in time, or one of the solutions.
- All solutions from a stored solution or current solution (see below).

Select the option to use from the **Solution at time**, **Eigenvalue**, or **Parameter value** lists (depending on the analysis type) in the **Initial value** and **Values of variables not solved for and linearization point** areas.

Using All Solutions from a Previous Analysis

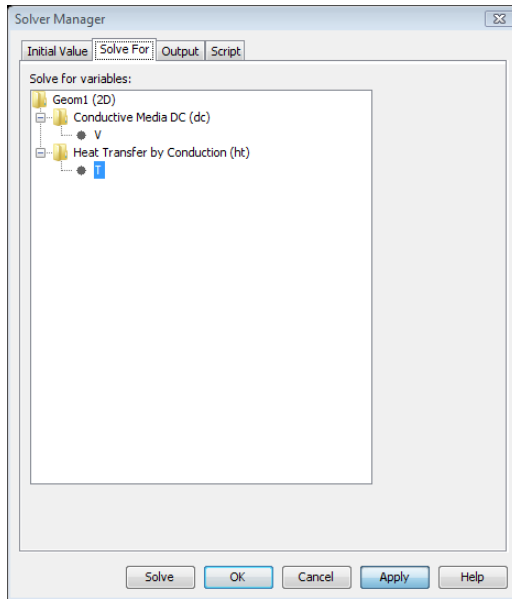
Selecting **All** in the **Solution at time** list in the **Values of variables not solved for and linearization point** area makes it possible to use all the solutions from a transient analysis as the value for the dependent variables not solved for in another transient analysis (the same applies to a parametric analysis). This method is effective when solving a one-way coupled time-dependent model. For example, start by solving for one dependent variable or application mode. Click the **Store Solution** button. Then click the **Stored solution** option button in the **Values of variables not solved for and linearization point** area and select **All** in the **Solution at time** list. Choose another variable or application mode on the **Solve For** page and click **Solve**. The values of the variables not solved for are now time-dependent and interpolated from the stored solution (because the second analysis uses different time steps from the first one). Alternatively, you can instead click the **Current solution** option button, which eliminates the need to click **Store Solution**. The drawback with this alternative is that the first solution disappears when the second analysis is finished, which can be annoying if the second analysis returns an incomplete solution due to some failure.

To provide the best possible data for the interpolation in the second analysis, store the time steps from the solver instead of the specified output times. To do so, click the **Time Stepping** tab in the **Solver Parameters** dialog box and select **Time steps from solver** in the **Times to store in output** list.

For an example of using this method to use solutions from a previous analysis, see “Peristaltic Pump” on page 429 of the *COMSOL Multiphysics Model Library*.

Choosing Variables for Which to Solve

Use the **Solve For** page to determine which dependent variables (solution components) to solve for. Select the application mode(s) to solve for in the list.



COMSOL Multiphysics solves for the temperature only in this multiphysics model. Solving for all components is the default behavior.

Click the plus sign to access the individual solution components for each application mode. The names of the solution components that appear are the names of the degrees of freedom (DOF) in the model. If you use Lagrange shape functions, these names are identical to those of the dependent variables. For other shape functions, there might be several DOF names corresponding to a dependent variable or several dependent variables corresponding to a DOF name.

The software takes the values of the solution components not solved for from results of a previous analysis or from the initial-value settings (see “Choosing the Values of Variables Not Solved For” on page 438). For the time-dependent and parametric solvers, these solution components can be varying with time or a parameter (see “Using All Solutions from a Previous Analysis” on page 439).

EXAMPLE—ORDERING OF EQUATIONS THAT YOU SOLVE SEQUENTIALLY

Consider the system of equations

$$\begin{aligned}\Delta u + v &= 0 \\ \Delta v &= 1\end{aligned}$$

where the variables u and v both are discretized with Lagrange elements. Then the corresponding degrees of freedom have the names u and v . Assume that you use a general form PDE application mode for the variables u and v (in this order) and that you enter the equations just mentioned in the displayed order. Then u corresponds to the first equation and v corresponds to the second one.

Assume that you solve only for u . Then the value of v is fixed to some given function, say $v = xy$. This means that COMSOL Multiphysics ignores the second equation and solves the single equation

$$\Delta u + xy = 0 .$$

If you instead enter the equations in the reverse order

$$\begin{aligned}\Delta v &= 1 \\ \Delta u + v &= 0\end{aligned}$$

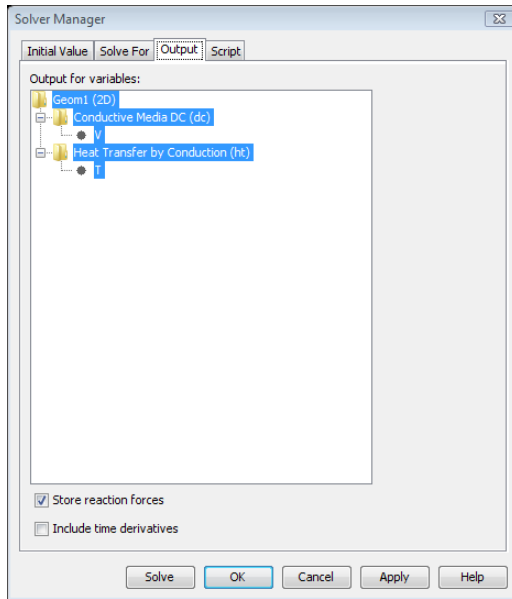
then again the software ignores the second equation with the result

$$\Delta(xy) = 1 ,$$

which is an equation with no solution. Thus some care is needed when choosing the order of variables and equations if you solve the equations sequentially using the settings on the **Solve For** page.

Selecting Variables for Solution Output

Use the settings on the **Output** page to determine which solution components to save for postprocessing. Click the plus sign in the list as shown in the following figure to access the individual dependent variables (or degrees of freedom; see “Choosing Variables for Which to Solve” on page 439) for each application mode.



The Output for variables list normally contains all dependent variables.

Normally the results are available for all solution components. However, it might be useful to reduce the size of the results data if you are not interested in postprocessing values for an application mode or solution component that takes part in the analysis. In postprocessing, the solution components not selected have the value 0.

By default, the solver computes the reaction force data from constraints and stores them so that they are available for postprocessing. For the time-dependent solver this computation might involve some extra assembly time for each time the solution is requested in the output. For this reason, you might want to disable this computation by clearing the **Store reaction forces** check box.

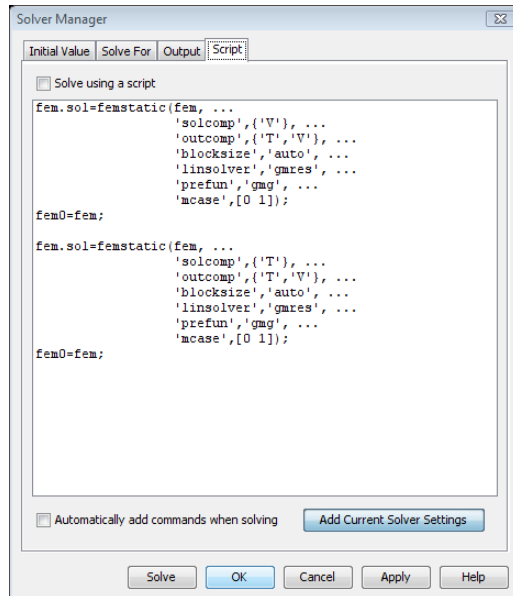
If you want to postprocess the time derivative of the solution components for a time-dependent analysis, select the **Include time derivatives** check box. The software then includes the first time derivative of the solution in the output. With this option, postprocessing at interpolated times becomes much more accurate because the software uses third-degree Hermite interpolation instead of the default linear interpolation. Without this option, postprocessing of second-order time derivatives gives 0.

Note: The size of the solution data doubles when you include the time derivatives.

Solver Scripting

With scripting it is possible to run the COMSOL Multiphysics solvers in a sequence, for instance, using the solution from one solver when calculating the initial value for the next. Scripts can automate the solution procedure for models that are difficult to get to converge without solving for only some of the variables at a time or where a sequential solver approach is more efficient, for example, for some multiphysics models with couplings in one direction only.

To access solver scripting, go to the **Solver Manager** dialog box and click the **Script** tab. It brings up a large text area where you can enter a sequence of commands using the solver commands `adaption`, `asseminit`, `femeig`, `femstatic`, `femlin`, `femlnlin`, and `femtime`. See the “Command Reference” chapter of the *COMSOL Multiphysics Reference Guide* for the syntax of these functions and details about available properties.



A solver script in the Solver Manager.

CREATING A SOLVER SCRIPT

Beside entering the script manually, you can generate it automatically in two ways:

- When you click the **Add Current Solver Settings** button, COMSOL Multiphysics automatically adds commands to the end of the script, commands corresponding to those it would use to solve the model at that moment. By changing settings between each click on this button you can generate a script without solving the model.
- If you select the **Automatically add commands when solving** check box, COMSOL Multiphysics adds commands to the end of the script each time you have it solve a model. Use this method to generate the script while manually solving a model using different settings, for example, on a coarse mesh to compute the solutions quickly.

Scripts can also include loops and branches. You could, for example, place `for i=1:10` at the beginning of the solver command sequence and `end` at its end. Any loop variable is valid in the `for` statement.

SOLVING USING A SOLVER SCRIPT

When you select the **Solve using a script** check box, COMSOL Multiphysics uses the script to compute the solution; otherwise, it uses the solver selected in the **Solver Parameters** dialog box.

The script runs in a context where `fem0` is an FEM structure with an extended mesh and a solution corresponding to the current solution in the user interface; `fem1` is the same except its solution corresponds to the stored solution. The FEM structure `fem` (or `xfem`, if multiple geometries are used) contains a newly created extended mesh. When the script has finished, the solution in `fem` is the one that COMSOL Multiphysics uses for postprocessing (the current solution).

Scripting is helpful for automating sequential solver steps where you first solve for one variable, then use that result to solve for another variable, and so on, or for switching between different solvers. To automatically run a solution and record the solver commands quickly, a good tip is to first use a coarse mesh and then re-solve using the script with desired mesh density.

Computing the Solution

Starting and Restarting the Solvers

There are several ways to start the solvers for an analysis:

- Click the **Solve** button on the Main toolbar, choose **Solve Problem** from the **Solve** menu, or click **Solve** in the **Solver Manager**. Any of these actions starts the analysis using the solver selected in the **Solver Parameters** dialog box. The solver runs according to settings in the **Solver Manager** dialog box (see “The Solver Manager and Solver Scripting” on page 436).
- Click the **Restart** button on the Main toolbar or choose **Restart** from the **Solve** menu. Either action restarts the analysis using the current solution as an initial value, the current values of variables not solved for, and the linearization point. In other words, the solver operates as if you had selected the **Current solution** option in the **Solver Manager** dialog box. Restarting the solution process is helpful, for example, for using a stationary solver to first compute a good initial value for a nonlinear or transient analysis, or for interrupting and continuing a transient simulation. For a transient analysis that is not time invariant (where there is an explicit dependence on time in the equations), note that you probably want to change the time list in the **Times** edit field before restarting the solver.

Updating a Model

Use the **Update Model** command on the **Solve** menu to update the model in the following situations:

- You have added or edited variables and want to use the modified variables during postprocessing without solving the model again.
- You have edited the equations, material properties, or boundary conditions and want to use the modified equations during postprocessing without solving the model again.
- You have changed the element type or element order and want to interpolate the current solution onto the new elements for postprocessing purposes.
- You have remeshed a geometry or modified the geometry and want to interpolate the current solution onto the new mesh for postprocessing purposes.

In all these cases, the software interpolates the current solution (or just passes it) to the Postprocessing mode. It does not recompute the solution to reflect any changes in variables, equations, mesh, and geometry.

Getting the Initial Value

The **Get Initial Value** command on the **Solve** menu evaluates the initial value for all solution components; it then passes the result to the Postprocessing mode so that it becomes the current solution. To compute the initial value, COMSOL Multiphysics uses the settings in the **Initial value** area in the **Solver Manager**. These settings apply to all solution components, even those the solver would not solve for. Therefore, for a time-dependent model, the result of the **Get Initial Value** command can sometimes differ from the solution at the initial time in the solution components not solved for.

The **Get Initial Value** command is useful when you want to:

- Plot and verify the initial conditions
- Postprocess the value of an expression without first solving the model
- Use a solution from a time-dependent or parametric analysis for further work

To do the latter, first store one of the solutions by clicking the **Store Solution** button on the **Initial Value** page of the **Solver Manager**, and then click the **Stored solution** option button in the **Initial value** area. You can then use the **Get Initial Value** command to make the current solution be this solution only. See “The Solver Manager and Solver Scripting” on page 436 for more information about the settings in the **Solver Manager** dialog box.

Note: The **Get Initial Value** command overwrites the current solution if one exists.

Solution Progress

When you solve a problem it is useful to know what is going on. For that purpose COMSOL Multiphysics provides a **Progress** window. It appears when you mesh a geometry, solve a problem, open a model or CAD file, or create a model report. At the top of the **Progress** window you can see what is going on at the moment. Just below is a progress bar displaying an estimate of the completed fraction of the total solving

process. In addition to progress information for the solvers, the **Progress** window provides

- Convergence plots (for the solvers that provide convergence information)
- Probe plots that show the current value of model quantities during solution (when you have defined such plots)
- A log of the solver commands

Using the mouse, you can resize the window to, for example, display more progress information or multiple plots.

CANCELING SOLVERS

Sometimes you might realize that you are solving the wrong problem, or perhaps it is taking too long; in such situations simply click the **Cancel** button. The solver attempts to cancel the computation as soon as possible and return the program to where it was before starting to solve.

KEEPING THE PROGRESS WINDOW OPEN

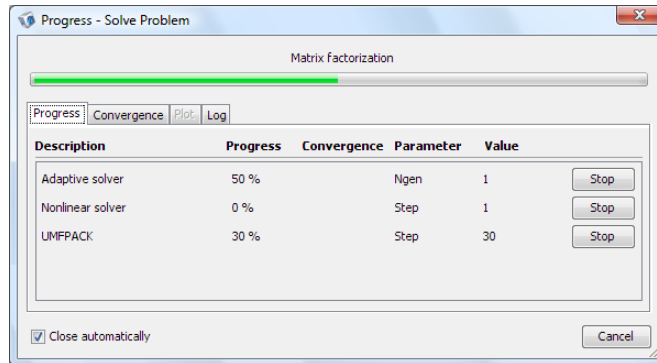
By default the **Progress** window disappears when the solver has finished successfully. To make the **Progress** window stay open when the solver has finished, clear the **Close automatically** check box. This option is useful, for example, for taking a closer look at the **Log** page. Note, though, that you can still examine the log when the **Progress** window is no longer open; go to the **Solve** menu and choose **View Log**.

If something goes wrong during the solution process, the **Progress** window does not close and the error message appears in red.

THE PROGRESS PAGE

The solvers call each other in a hierarchical order: the adaptive solver calls the linear, nonlinear, parametric, or eigenvalue solver; the parametric solver calls the nonlinear or linear solver; the time-dependent, eigenvalue, linear, and nonlinear solvers all call both

the assembly and the linear system solver. The solver hierarchy is visible on the **Progress** page because each solver introduces its own line when it is called.



The Progress window showing the progress for the adaptive and nonlinear solvers.

The **Description** column shows the solver's name, the **Progress** column shows an estimate of the solver's progress, and the **Convergence** column shows an estimate of the solver's convergence if available. The **Parameter** and **Value** columns contain solver-dependent information: the adaptive solver shows the adaptive generation number; the time-dependent solver shows the time; the parametric solver shows the parameter value; the nonlinear solver and iterative linear system solvers show the iteration number.

Using the Stop Buttons

You can use the **Stop** buttons that appear at each solver level to stop the solver's execution. When you click a **Stop** button, COMSOL Multiphysics returns a current approximation to the solution if possible. For example, when you press it at the adaptive solver level, the underlying linear, nonlinear, parametric, or eigenvalue solver continues until it is finished, but the adaptive solver stops at its current generation, immediately returning a solution for postprocessing. Similarly, you can click the **Stop** button to return the current iteration for the nonlinear solver or an iterative solver. Also use the **Stop** button during time stepping to return all time steps up to the current one. The parametric solver works similarly: to return the solutions for all parameter values up to the current one, click the **Stop** button.

THE CONVERGENCE PAGE

Click the **Convergence** tab to see a convergence plot showing an error estimate against the iteration number for the nonlinear solver and for the iterative linear system solvers (the Conjugate gradients, BiCGStab, GMRES, FGMRES, and multigrid solvers).

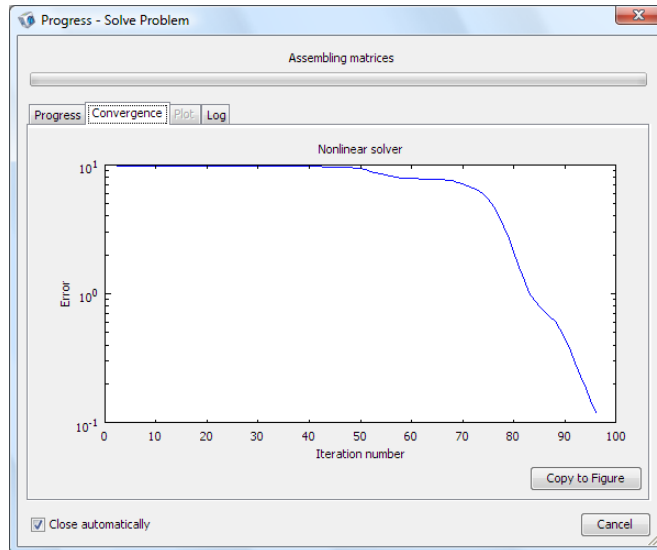
For the nonlinear solver, the convergence plot shows the relative error estimate (see “Nonlinear Solver Settings” on page 389) for each nonlinear iteration number. Note that this number also appears on the **Log** page as **ErrEst**.

For the iterative linear system solvers, the error estimate for each linear iteration is a factor times the relative (preconditioned) residual (see “Convergence Criteria” on page 550 of the *COMSOL Multiphysics Reference Guide*). This number also appears on the **Log** page as **LinErr**. When these solvers are used together with the nonlinear solver, the graphs for the different linear-system solution steps are merged, and the plots use the accumulated number of iterations.

When using the parametric solver, the graphs for the different parameter steps are merged, and the plots use the accumulated number of iterations.

When using a time-dependent solver, the graph on the **Convergence** page shows the reciprocal of the time step size versus the time step. That is, a convergence plot with decreasing values shows that the time-dependent solver takes longer time steps, and vice versa.

Note that the error estimate numbers for the last iteration also appear in the **Convergence** column on the **Progress** page.



The Progress window's Convergence page shows a convergence plot for the nonlinear solver.

Click the **Copy to Figure** button to create a copy of the convergence plot in a separate figure window.

THE PLOT PAGE

Click the **Plot** tab to see any *probe plots* that you have defined to monitor aspects of the solution. For more information, see “Plotting Values While Solving” on page 481.

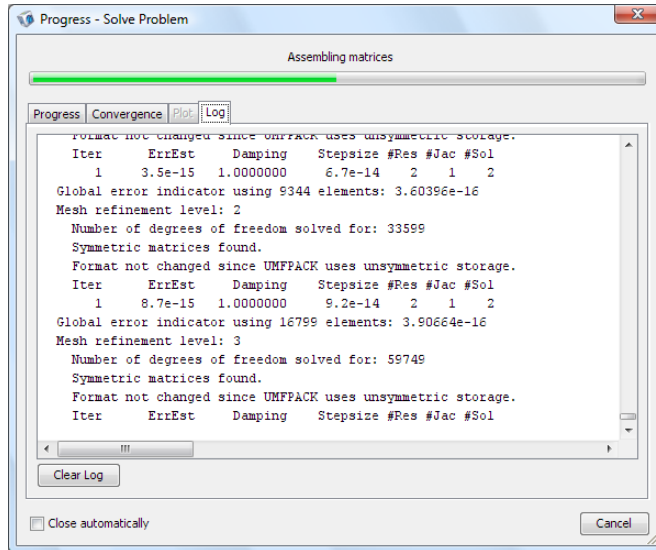
THE LOG PAGE

The **Log** page contains logs from the solvers. When a solver starts, the log window displays the solver call using COMSOL Multiphysics command-line syntax (see the “Command Reference” chapter of the *COMSOL Multiphysics Reference Guide* for more information), and the number of degrees of freedom solved for. This page contains logs from all solver calls during a session. Click the **Clear Log** button to start with a fresh log page.

Note: Clicking the **Clear Log** button only clears the log page, not any log files that you may be using. (See “Logging to a File” on page 452.) To clear the log file, choose **Options>Preferences** and click the **Clear Log File** button.

The Adaptive and Parametric Solver Logs

The adaptive solver prints a section in the log for each adaptive generation containing the current number of elements and an error indicator. The parametric solver similarly outputs one section to the log for each parameter value.



The Progress window showing the log from the adaptive and nonlinear solvers.

The stationary, time-dependent, and eigenvalue solvers log their iterations.

The Nonlinear Solver Log

The log from the nonlinear solver contains the following information:

- The iteration number (**Iter**)
- A relative error estimate (**ErrEst**) (see “Nonlinear Solver Settings” on page 389)
- The damping factor used in each Newton step (**Damping**)
- The size of the undamped Newton step (**Stepsize**) in the error estimate norm
- The numbers of residuals (**#Res**), Jacobians (**#Jac**), and linear-system solutions computed (**#Sol**) so far

The Iterative Linear System Solver Logs

The iterative linear system solvers produce a log that additionally contains the total number of linear iterations (**LinIt**), a relative error estimate (**LinErr**), and the relative residual (**LinRes**). The relative error estimate is a factor times the relative

(preconditioned) residual (see “Convergence Criteria” on page 550 of the *COMSOL Multiphysics Reference Guide*). The relative residual is the Euclidean norm of the residual divided by the norm, $|b|$, of the linear system’s right-hand side.

The Time-Dependent Solver Log

The time-stepping algorithm produces a log that contains

- The time step number (**Step**)
- Time (**Time**; output times are indicated with **out**)
- The step size (**Stepsize**)
- The number of residuals (**Res**), Jacobians (**Jac**), and linear system solutions (**Sol**) computed

You can also see the order of accuracy of the method (**Order**), the number of error test failures in time stepping (**Tfail**), and the number of failures in the nonlinear iterations (**NLfail**). For iterative linear system solvers, the log also contains the total number of linear iterations, a linear error estimate, and a relative residual (see above).

The Eigenvalue Solver Log

The eigenvalue solver produces a log that contains the iteration number (**Iter**), an error estimate (**ErrEst**), the number of converged eigenvalues (**Nconv**), and—if you are using an iterative linear solver—the number of linear iterations (**LinIt**).

Logging to a File

You can also direct the log from the **Log** page to a file. This is useful if the solver log becomes very long, or if you want to keep the log for further analysis or comparison. To activate logging to file, choose **Preferences** from the **Options** menu and select the **Log to file** check box. Choose a log file by typing in the path in the **Log file** edit field or browse to a file by clicking the **Browse** button. The file logging appends the log messages to any already existing content of the log file. To start with a new empty log file, click the **Clear Log File** button.

Monitoring Memory Usage

When running COMSOL Multiphysics on Windows, you can inspect the memory usage at any time during a modeling session in the **Memory** field at the bottom right of the user interface. This field displays the numbers in MB for the current and peak virtual memory usage, separated by a slash. When the main user interface window is in focus and tooltip display is active, placing the cursor over the **Memory** field additionally shows the current and peak physical memory usage.

Postprocessing and Visualization

To help you analyze results from its solvers, COMSOL Multiphysics provides numerous postprocessing and visualization tools, including advanced graphics, data display and export functions, and a report generator. The following chapter gives details on how to use these tools.

Postprocessing Results

COMSOL Multiphysics Postprocessing Overview

COMSOL Multiphysics provides many tools for postprocessing and visualizing model quantities. It creates a wide variety of plots:

- Surface plots
- Slice plots
- Isosurface plots
- Contour plots
- Streamline plots
- Pathline plots that describe particle traces
- Principal stress/strain plots
- Combinations of these plots

In addition, animations reveal the dynamics in a solution. Cross-section plots can interpolate within a domain to extract mesh-independent plots, and the software can also extend these plots in time or along parametric solutions. Domain plots provide similar capabilities for geometric domains. Global plots can visualize globally defined variables, typically solutions to ODEs and algebraic equations. You can also compute numerical integrals of model quantities as well as display and save postprocessing data.

Postprocessing Expressions

COMSOL Multiphysics does not limit postprocessing to predefined quantities; you can postprocess any function by entering the corresponding expression. The software can postprocess expressions that combine user-defined variables, application mode variables, standard variables, mathematical and logical functions (see “Using Mathematical and Logical Operators and Functions” on page 147), and numbers. Enter the expression that you want to plot or evaluate in the **Expression** edit field in the dialog box for the type of postprocessing that you are interested in. This edit field is available for most plot types as well as for integration and data display and evaluation. Pressing Ctrl+Space opens the hierarchical list of predefined quantities that you also find in the corresponding **Predefined quantities** list. The difference between the two ways of accessing the list is that pressing Ctrl+Space and then selecting a quantity from the list adds the name of the corresponding variable to the **Expression** edit field at the

current location instead of replacing any previous content (for example, pressing Ctrl+Space and selecting **Plane Stress (ps)>Total displacement** appends `disp_ps` to the expression in the **Expression** field). This makes it easier to build an expression with valid variable names.

ACCESSING PREDEFINED VARIABLES

The *COMSOL Multiphysics Modeling Guide* and the documentation for the modules provide lists of application mode variables. For a list of standard variables of this type, see “Variables and Expressions” on page 138.

ACCESSING OTHER SOLUTIONS THAN THE SELECTED SOLUTION

When you use the names of the dependent variables in a postprocessing expression, COMSOL Multiphysics uses the solution associated with the selected parameter value, eigenvalue, or time for a parametric analysis, eigenvalue analysis, or time-dependent analysis, respectively. To access other solutions in the model, use the `with` operator. For more information about how to use this operator, see “The with Operator” on page 166.

EVALUATION OF UNDEFINED QUANTITIES

During the evaluation of expressions, by default COMSOL Multiphysics does not report partially undefined quantities, and the program plots a quantity where it is defined. If a postprocessing quantity is undefined everywhere, an error occurs for all plot types except arrow plots. If you select the option **Give error for undefined operation or variable** described on page 487, an error occurs whenever an expression is not defined everywhere, also for arrow plots. For example, in a plot of the Lagrange multiplier on the boundary using weak constraints, that multiplier is undefined on unconstrained boundaries. Following the rule above, the software plots the Lagrange multiplier on constrained boundaries if at least one exists, but if you have selected the option **Give error for undefined operation or variable**, an error message is given.

Selection of Frames for Postprocessing

When a model works with deformed meshes, the **Frame** list is enabled in all postprocessing dialog boxes, and it contains the names of all available frames. There you select which frame to use for the desired postprocessing task. For further information about frames and deformed meshes, see the section “Frames for Deformed Meshes” on page 458 of the *COMSOL Multiphysics Modeling Guide*.

Interpolating Expressions at Arbitrary Points

The point cross-section plot makes it easy to view an expression at an arbitrary set of spatial coordinates and results in a line plot. To get the value of a certain subdomain expression or variable at any location in the geometry, use the **Data Display** dialog box. To get the value of expressions defined at a geometry vertex, use the **Point Evaluation** dialog box (see “Getting Numerical Results Directly” on page 459). Note that expressions and variables that include derivatives of the dependent variables (for example, stresses in a structural analysis) are not available at isolated geometry vertices (points). Using **Export>Postprocessing Data** on the **File** menu, COMSOL Multiphysics prints the values to a file (see “Exporting Postprocessing Data to a File” on page 487).

Computing Quantities Using Integration

You can integrate any postprocessing quantity over subdomains and boundaries to compute quantities such as total flux, charges, inductances, reaction forces, and average values.

- To integrate over subdomains in 1D, 2D, and 3D geometries, use the **Subdomain Integration** dialog box.
- To integrate over boundaries 2D and 3D geometries, use the **Boundary Integration** dialog box.
- To integrate over edges in 3D geometries, use the **Edge Integration** dialog box.

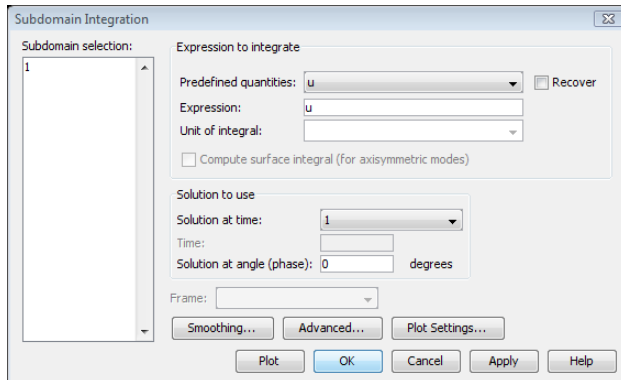


Figure 7-1: The Subdomain Integration dialog box.

To integrate a quantity:

- 1 Open the integration dialog box for the domain type over which you want to integrate a quantity.
- 2 Select the domains over which to integrate in the list of subdomains, boundaries, or edges.
- 3 Make an entry in the **Expression to integrate** edit field. Select from application-mode dependent quantities in the **Predefined quantities** list or enter an expression containing variables. Select the **Recover** check box to use accurate derivative recovery (see “Accurate Derivative Recovery” on page 469).
- 4 For an eigenvalue, parametric, or time-dependent problem, go to the list of available solutions in the **Solution to use** area and make an appropriate selection to use for the integration. For time-dependent problems, you can select **Interpolated** from the list, and enter an arbitrary time in the **Time** edit field.
- 5 Click **Apply** or **OK** to evaluate the integral and display the result in the message log.

Use the **Solution at angle (phase)** option to integrate complex-valued expressions by multiplying the solution by a factor of $\exp(i*\pi*phase/180)$ prior to expression evaluation.

For 2D axisymmetry models, the option **Compute volume integral** in the **Subdomain Integration** dialog box and the option **Compute surface integral** in the **Boundary Integration** dialog box is enabled. If you select either of these check boxes, the software multiplies the expression (integrand) with $2*\pi*r$ prior to integration to compute the corresponding volume or surface integral.

INTEGRATION ORDER, INTEGRATION TYPE, AND SMOOTHING

To change the integration order and integration type, click the **Advanced** button. In the **Advanced Settings** dialog box that opens, you can choose an integration method from the **Method** list:

- **Automatic** is the default setting, which means that the integration method is the standard numerical integration method for most properties. Predefined quantities for reaction forces use summation instead of integration. Also, if you specify an expression for the integrand using the `reacf` operator, the automatic setting chooses the summation method. You can read more about the `reacf` operator in the section “Using Special Operators” on page 163.
- **Integration**—this is the standard numerical integration method (quadrature).
- **Summation**—this is a summation method that is useful for computing reaction forces. The summation method finds all nodes on the boundary, evaluates the expression in the nodes and sum up the values. Reaction force variables are

predefined in the structural mechanics application modes. Read more about reaction forces in the *Structural Mechanics Module User's Guide* in the chapter “Calculating Reaction Forces” on page 113.

For the integration order, the **Auto** check box is selected by default. The software then automatically determines an appropriate integration order for the expression. Clear the **Auto** check box to make an entry in the **Integration order** edit field. The software performs the integration elementwise using numeric quadrature of the selected order.

If you have selected the **Recover** check box, you can specify how the accurate derivative recovery handles borders between domains with different settings. To do this, click the **Smoothing** button to open the **Smooth Parameters** dialog box. Click the **Smooth everywhere** button, which means smoothing in the entire geometry, or click the **Smooth only in domains with equal settings** button (the default), which means that no smoothing takes place across borders between domains with different settings, which often correspond to material discontinuities.

PLOTTING VALUES OF INTEGRALS AS A FUNCTION OF TIME OR A PARAMETER

For a time-dependent or parametric solution, a **Plot** button and a **Plot Settings** button are available. Clicking the **Plot** button creates a plot of the computed value as a function of time or of the parameter. Clicking the **Plot Settings** button opens the **Plot Settings** dialog box where you can specify plot properties.

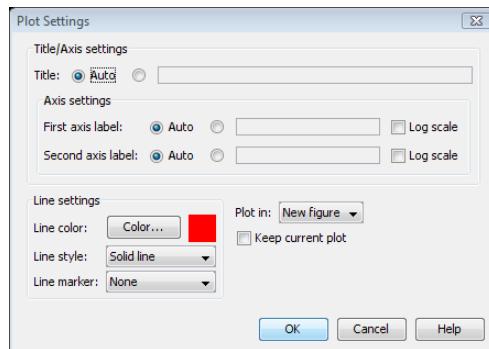


Figure 7-2: The Plot Settings dialog box.

In the **Line settings** area, specify the color, line style, and marker of the plotted line. In the **Title/Axis settings** area, specify the title and axis labels of the plot. If **Auto** is selected, the plot uses an automatically generated title. Select the **Log scale** check boxes to use logarithmic scales on the axes in the plot. From the **Plot in** list, select in which figure

the plot will appear. To visualize multiple data in the same figure, select the **Keep current plot** check box.

INTEGRATION OF UNDEFINED QUANTITIES

Integration works differently than evaluation. When COMSOL Multiphysics attempts to integrate an expression over a domain where it is not defined, an error occurs.

Getting Numerical Results Directly

The Postprocessing mode in 2D can display the numerical value of the current surface or contour plot. Get this value by clicking anywhere in the model domain. The message log shows the value of the plotted property. In 1D, click in the plot area to display the value of the currently plotted expression. If snapping is active, the selected points snap to the grid points.

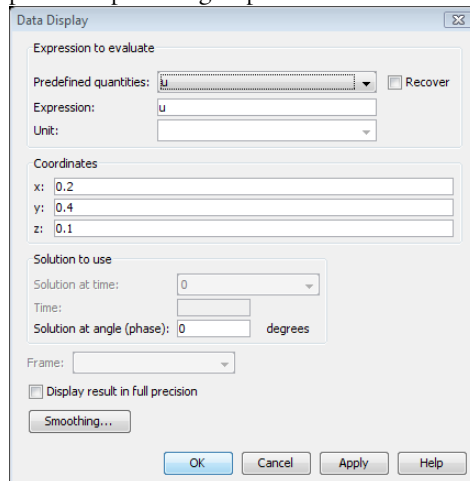


Figure 7-3: The Data Display dialog box.

Use the **Data Display** dialog box to evaluate a subdomain expression at a certain coordinate. If applicable, select the solution to use and then select one of the predefined quantities or enter an arbitrary expression containing variables. Select the **Recover** check box to use accurate derivative recovery (see “Accurate Derivative Recovery” on page 469). If you want to evaluate the expression at several coordinates you can enter them separated by spaces in the edit fields in the **Coordinates** area.

Click **Apply** or **OK** to display the value of the expression in the message log at the bottom of the user interface. Select the **Display result in full precision** check box to get

the value with as many significant digits as possible. To control the setting of this check box for new models, use the corresponding check box on the **Postprocessing** page in the **Preferences** dialog box.

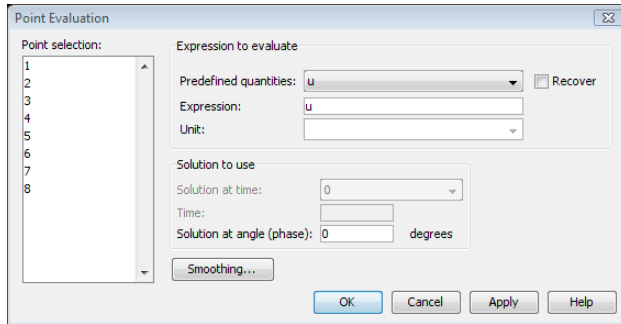


Figure 7-4: The Point Evaluation dialog box.

Use the **Point Evaluation** dialog box to evaluate expressions defined in a geometry vertex. This dialog box works the same as the integration dialog boxes described in “Computing Quantities Using Integration” on page 456, but for vertices rather than subdomains, boundaries, and edges.

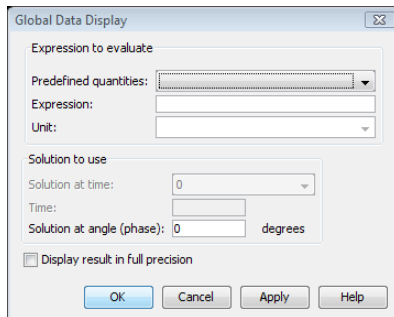


Figure 7-5: The Global Data Display dialog box.

To get the value of a globally defined quantity, use the **Global Data Display** dialog box. In the **Expression to evaluate** area choose the quantity to display. Either select from predefined quantities or enter an expression containing variables. If applicable, select the solution to use. Click **Apply** or **OK** to display the value of the expression in the message log at the bottom of the user interface.

Plotting Global Expressions

To visualize globally defined variables, such as solutions to ODEs or coupling variables with global destination, use the **Global Variables Plot** dialog box.

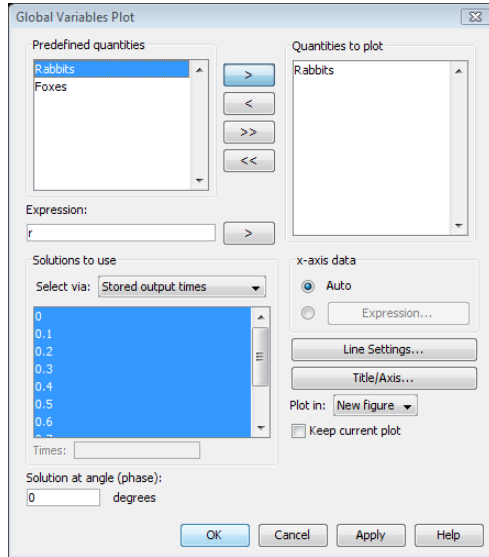


Figure 7-6: The Global Variables Plot dialog box.

In the **Predefined quantities** list, the names of all variables from ODEs and other global equations (defined in the **Global Equations** dialog box) and global expressions (defined in the **Global Expressions** dialog box) are present. To plot one or more of these, select them and click the **>** button to add them to the **Quantities to plot** list. When you select a name in the **Predefined quantities** list, the software displays its variable name in the text field below the list. You can edit this name and add the edited quantity using the **>** button to the right of the text field. To remove one or more expressions from the **Quantities to plot** list, select them and click the **<** button. You can use Shift-click and Control-click in both lists.

In the **Solutions to use** area, select which time steps to include in the plot. By choosing **Interpolated times** from the **Select via** list, you can enter any vector of times in the **Times** edit field, such as 0:0.02:1. This way of choosing the times for plotting is also available in the **Cross-Section Plot Parameters** and **Domain Plot Parameters** dialog boxes when postprocessing a time-dependent model.

Control the quantity to plot on the x -axis in the **x-axis data** area. If **Auto** is selected, the x -axis corresponds to time.

The settings that you access by clicking the **Line Settings** button are described under “Line Settings” on page 477, and the settings that you specify by clicking the **Title/Axis** button are described under “Adjusting the Plot Title and Axis” on page 466. The settings in the **Plot in** list are described under “Plotting Data to Separate Figure Windows” on page 465, and the **Solution at angle (phase)** edit field is described under “Visualizing Complex-Valued Quantities” on page 464.

Visualization Using the Plot Parameters Dialog Box

To get visualizations on the solution domain using the common plot types, use the **Plot Parameters** dialog box. The choices include:

- For all geometries: plot of the geometry edges
- In 1D: line plot
- In 2D: surface, contour, arrow, deformed shape, streamline, particle-tracing, and line plots
- In 3D: slice, isosurface, subdomain, boundary, edge, arrow, deformed shape, particle-tracing, and streamline plots

In addition, the choice of plotting principal stress/strain plots is available with the application modes for structural mechanics in the Structural Mechanics Module, MEMS Module, and Acoustics Module.

To open the **Plot Parameters** dialog box, either click the **Plot Parameters** button on the Main toolbar or choose **Plot Parameters** from the **Postprocessing** menu.

Use the **General** page to set common plot parameters such as the plot types.

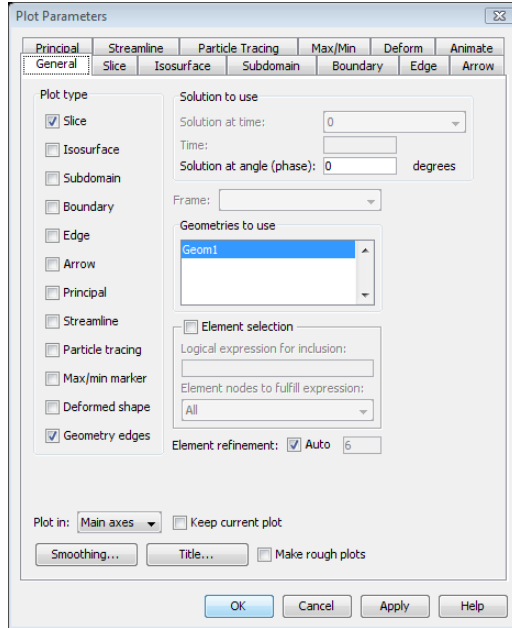


Figure 7-7: The 3D Plot Parameters dialog box.

SELECTING A PLOT TYPE

In the **Plot type** area you indicate which types to use in the plot. The **Geometry edges** check box is normally selected to show the geometry edges; clear this check box to exclude geometry edges, for example, when using line plots on geometry edges.

For details on the various plot types see “1D Standard Plots” on page 497, “2D Standard Plots” on page 502, and “3D Standard Plots” on page 534.

SELECTING A SOLUTION

For an eigenvalue, parametric, or time-dependent problem, select which solutions to use for the visualization in the corresponding list. For time-dependent problems, you can select **Interpolated** from the list and then enter an arbitrary time within the simulation time frame in the **Time** edit field.

SELECTING GEOMETRIES

For a problem using multiple geometries, select which geometries to include in the plot in the **Geometries to use** list. In this list, only the geometries with the same space

dimension and same names of the independent variables as the current geometry are displayed. Note that the selected plots are only rendered where the corresponding expression is defined. This means that the plotted expressions must be defined on all of the selected geometries in order for the plot to be rendered on all selected geometries. For an example of the use of this feature, see “Example—Thin-Layer Diffusion” on page 442 in the *COMSOL Multiphysics Modeling Guide*.

VISUALIZING COMPLEX-VALUED QUANTITIES

All plot types show the real part of a complex-valued quantity. You can visualize complex-valued expressions by multiplying the solution with a factor of $\exp(i\pi \cdot \text{phase}/180)$ prior to expression evaluation. To do so, enter a phase value (in degrees) in the **Solution at angle (phase)** edit field. See also “Complex-Valued Functions” on page 487.

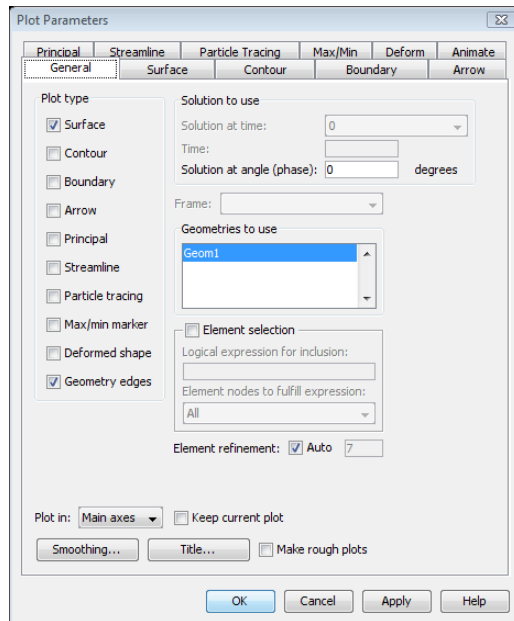


Figure 7-8: The 2D Plot Parameters dialog box.

CONTROLLING ELEMENT REFINEMENT

The value in the **Element refinement** edit field helps determine the quality of the plot. By resolving higher-order elements into several subelements, the visualization of colors within the higher-order elements uses linear interpolation of colors within each subelement. The element refinement also provides an approximation of the curved

mesh elements by inserting additional nodes between the nodes in the mesh. Even if a model only contains linear elements, a higher refinement might increase the quality of the plot, at least when the model contains large elements, due to the color interpolation method used on the graphics card. The software refines the elements in a uniform regular pattern. A higher refinement is necessary for models with few elements, whereas you can use a lower refinement when viewing a model with many elements. Higher refinement provides a smoother plot but requires more memory. Determine the refinement by specifying an integer number that indicates the refinement steps. The following list shows the factor that the number of elements grows with for the first three steps:

REFINEMENT STEP	1D	2D	3D
1	1	1	1
2	2	4	8
3	3	9	27

Selecting the **Auto** check box makes this choice automatically, by internally computing a suitable refinement based on the maximum element order and the total number of elements evaluated on. This value then automatically appears in the **Element refinement** edit field. If you combine several plot types, the automatically entered value is the maximum refinement value for the plot types.

USING ELEMENT SELECTION

It is possible to control which elements that appear in a plot. To do so, select the **Element selection** check box and make an entry in the **Logical expression for inclusion** edit field such as $x > 0 \ \& \ (y < -5 \ | \ y > 5)$. A setting in the **Element nodes to fulfill expression** list instructs the software to plot the volume elements of the mesh that fulfill these conditions: all nodes; at least one node; or at least one but not all nodes.

On the **Subdomain** page in 3D models, use the **Fraction of elements to plot** edit field to specify how many volume elements to plot (as a fraction between 0 and 1). COMSOL Multiphysics selects the fraction based on user settings in the **Select elements** list. Choose between keeping the elements with smallest or greatest values or a random selection (selecting randomly is the default setting). These settings only apply to subdomain plots.

PLOTTING DATA TO SEPARATE FIGURE WINDOWS

Standard plots appear in the main drawing area. You can also plot results in a separate COMSOL figure window by selecting **New figure** or an existing figure window in the

Plot in list. This figure window can be accessed from the COMSOL Script prompt. For further details, see “The Figure Window” on page 551.

When using COMSOL Multiphysics with MATLAB, you can select **MATLAB figure** in the **Plot in list**. This option creates the plot in a MATLAB figure window, similar to what the `postplot` command would create. With this selection, you cannot use the **Keep current plot** check box. To plot in an existing figure window, type `hold on` at the command prompt.

OVERLAYING MULTIPLE PLOTS

Select the **Keep current plot** check box to generate multiple overlaid plots in the drawing area or a single separate figure window.

ADJUSTING THE PLOT TITLE AND AXIS

Click the **Title** button to open a dialog box where you can edit the plot’s title. In the **Cross-Section Plot Parameters**, **Domain Plot Parameters**, and **Global Variables Plot** dialog boxes, click the **Title/Axis** button to open a dialog box where you can edit the plot’s title and axis labels and select logarithmic scale on the plot axes for line and point plots.

Support for Formatting and Symbols in Texts

For the titles as well as the *x*-axis, *y*-axis, and *z*-axis labels, you can use formatted strings that include HTML tags, Greek letters, and mathematical symbols.

You can use the following HTML tags in the text strings:

TABLE 7-1: VALID HTML TAGS

HTML TAG	DESCRIPTION
 	Enclosed text is rendered using a bold font.
 	Line break.
<CENTER> </CENTER>	Centered text.
<I> </I>	Enclosed text is rendered using an italic font.
	List item. When the list used is (an ordered list) the LI element is rendered with a number. When the list used is (an unordered list) the LI element is rendered with a bullet.
 	Ordered list (see also:).
<P> </P>	Paragraph. This tag creates a line break and a space between lines.
<PRE> </PRE>	Enclosed text preserves spaces and line breaks. The text is rendered using a monospaced font.

TABLE 7-1: VALID HTML TAGS

HTML TAG	DESCRIPTION
<STRIKE> </STRIKE>	Enclosed text is rendered with a strike-through appearance.
	Enclosed text is rendered in subscript with the enclosed text slightly lower than the surrounding text.
	Enclosed text is rendered in superscript with the enclosed text slightly higher than the surrounding text.
<TT> </TT>	Enclosed text is rendered using a monospaced font.
<U> </U>	Enclosed text will be underlined.
 	Unordered list (see also:).

The texts in labels and titles support the following Greek character tags:

TABLE 7-2: VALID GREEK SYMBOL TAGS

TAG	SYMBOL	TAG	SYMBOL
\ALPHA	A	\alpha	α
\BETA	B	\beta	β
\GAMMA	Γ	\gamma	γ
\DELTA	Δ	\delta	δ
\EPSILON	E	\epsilon	ε
\ZETA	Z	\zeta	ζ
\ETA	H	\eta	η
\THETA	Θ	\theta	θ
\IOTA	I	\iota	ι
\KAPPA	K	\kappa	κ
\LAMBDA	Λ	\lambda	λ
\MU	M	\mu	μ
\NU	N	\nu	ν
\XI	Ξ	\xi	ξ
\OMICRON	O	\omicron	ο
\PI	Π	\pi	π
\RHO	P	\rho	ρ
\SIGMA	Σ	\sigma	σ
\TAU	T	\tau	τ
\UPSILON	Υ	\upsilon	υ

TABLE 7-2: VALID GREEK SYMBOL TAGS

TAG	SYMBOL	TAG	SYMBOL
\PHI	Φ	\phi	ϕ
\CHI	χ	\chi	χ
\PSI	Ψ	\psi	ψ
\OMEGA	Ω	\omega	ω

For texts in titles and axis labels, you can use the following mathematical symbol tags:

TABLE 7-3: VALID MATHEMATICAL SYMBOL TAGS

TAG	SYMBOL	TAG	SYMBOL
\approx	\approx	\bullet	\bullet
\sim	\sim	\partial	∂
\prop	\propto	\nabla	∇
\neq	\neq	\prod	\prod
\equiv	\equiv	\sum	\sum
\leq	\leq	\sqrt	$\sqrt{\quad}$
\geq	\geq	\int	\int
\gg	\gg	\oplus	\oplus
\gg	\ll	\otimes	\otimes
\plusminus	\pm	\larrow	\leftarrow
\infinity	∞	\rarrow	\rightarrow
\deg	$^\circ$	\leftrightarrow	\leftrightarrow
\cdot	\cdot	\ldarrow	\leftleftarrows
\times	\times	\rdarrow	\rightrightarrows

In addition to ASCII characters, Greek letters, and the mathematical symbols listed in Table 7-3, COMSOL Multiphysics correctly displays any Unicode-based character that you paste into a title or label edit field.

SETTING THE RANGE FOR THE DATA

For most plot types, the default for color mapping is to spread the range of colors across the entire range of variable values. Thus, in any given limited range of values it might be difficult to discern any differences. To overcome this effect, you can set the display to have the full range of colors map onto a subset of the variable values. To do so, click the **Range** button and in the resulting dialog boxes, clear the **Auto** check box and then enter a range in the **Min** and **Max** edit fields. In this case, parts of the plot outside the range are not rendered.

MAKING CONTINUOUS PLOTS OF DISCONTINUOUS DATA

By default, COMSOL Multiphysics plots enforce continuity on discontinuous data. The plot then does not reveal discontinuities in the results data that could depend on discontinuous material properties or the numerical method. To show the discontinuities, clear the **Smooth** check box for the specific data.

By default, the smoothing is applied within each group of domains with equal settings. Thus, there is no smoothing across material discontinuities. To get smoothing also over such discontinuities, click the **Smoothing** button to open the **Smooth Parameters** dialog box and select the **Smooth everywhere** button.

ACCURATE DERIVATIVE RECOVERY

Postprocessing of stresses or fluxes boils down to evaluating space derivatives of the dependent variables. By default, postprocessing of a derivative like u_x or u_{xx} (first and second derivatives of u with respect to x) is done by evaluating the derivative of the shape functions used in the finite element approximation. These values have poorer accuracy than the solution u itself. For example, $u_{xx}=0$ if u has linear elements. To get better accuracy, select the **Recover** check box. COMSOL Multiphysics then evaluates the derivatives (and u itself) using a polynomial-preserving recovery technique by Z. Zhang (see Ref. 1). The recovery is only applied on variables that are discretized using Lagrange shape functions. Because the recovery makes the derivatives continuous, there is usually no need to select the **Smooth** check box when using recovery.

The polynomial-preserving recovery is a variant of the superconvergent patch recovery by Zienkiewicz and Zhu that forms a higher-order approximation of the solution on a patch of mesh elements around each mesh vertex. For regular meshes, the convergence rate of the recovered gradient is $O(h^{p+1})$ —the same as for the solution itself. However, near boundaries, the accuracy is not as good, and it might even be worse than without recovery. Postprocessing is about 2–5 times slower when using accurate derivative recovery. For this reason, prefer using recovery of stresses or fluxes for surface plots, contour plots, slice plots, and isosurface plots. Use recovery to a lesser extent for subdomain plots, boundary plots, and edge plots.

By default, the accurate derivative recovery smoothes the derivatives within each group of domains with equal settings. Thus, there is no smoothing across material discontinuities. To get smoothing also over such discontinuities, click the **Smoothing** button to open the **Smooth Parameters** dialog box and click the **Smooth everywhere** button.

Selecting the **Recover** check box is equivalent to applying the `pprint` or `ppr` operator on the expression (see “The `ppr` and `pprint` Operators” on page 167). The `pprint`

operator corresponds to **Smooth only in domains with equal settings**, and the ppr operator corresponds to **Smooth everywhere** in the **Smooth Parameters** dialog box.

Model Example

For an example of how accurate derivative recovery helps to make the stress field smoother, see “Stresses in a Pulley” on page 577 of the *COMSOL Multiphysics Model Library*.

Reference

1. A. Naga and Z. Zhang, “The polynomial-preserving recovery for higher order finite element methods in 2D and 3D,” *Discrete and continuous dynamical systems—series B*, vol. 5, pp. 769–798, 2005.

SELECTING A COLORMAP

For many plot types you can select the colormap to use for coloring the surfaces, boundaries, contours, streamlines, slices, and so on. The following colormaps are available from the **Colormap** lists in the **Plot Parameters** dialog box:

- **jet**—The color ordering in this map corresponds to the wavelengths of the visible part of the electromagnetic spectrum: beginning at the small-wavelength end with dark blue, the colors range through shades of blue, cyan, green, yellow, and red. While colorful, this colormap has the disadvantage that it uses distinctions between reds and greens that individuals with the most common forms of color deficiency (affecting up to 10% of a technical audience) cannot perceive. (This colormap takes its name from an astrophysical fluid-jet simulation from the National Center for Supercomputer Applications, USA.)
- **hot**—This colormap ranges from black through red and yellow to white, which corresponds to the colors iron takes as it heats up.
- **hsv**—A colormap that varies the hue component of the hue-saturation-value color model, keeping the saturation and value constant (equal to 1). The colors begin with red, then pass through yellow, green, cyan, blue, magenta, and finally return to red. This map is particularly useful for displaying periodic functions and has a sharp color gradient.
- **wave**—This colormap is particularly useful for data that naturally has a positive and negative attribute in addition to a magnitude. An example of a *double-ended scheme*, it ranges linearly from blue to white, and then linearly from white to red. When the range of the visualized quantity is symmetric around zero, the color red or blue indicates whether the value is positive or negative, and the saturation indicates the magnitude. This color palette has the additional feature of being

interpretable by color-deficient users. Because it uses no red-green-gray distinctions, the palette is efficient for 99.98% of the population.

- **gray**—A colormap that uses no color, only the linear gray scale—the easiest palette to understand and order. Gray-scale plots are often much easier for a publisher to include in journal articles. Further, humans can perceive very fine structural detail in a gray scale that would go unnoticed in color. Using a gray scale also raises the probability that a plot is correctly interpreted by individuals with certain visual disabilities. This map varies linearly from black (1, 1, 1) to white (0, 0, 0).
- **grayprint**—Varies linearly from dark gray (0.95, 0.95, 0.95) to light gray (0.05, 0.05, 0.05). This colormap overcomes two disadvantages that the standard gray colormap has when used for printouts on paper, namely that it gives the impression of being dominated by dark colors, and that white cannot be distinguished from the background.

All colormaps are also available for surface plots generated using the **Cross-Section Plot Parameters** and **Domain Plot Parameters** dialog boxes. To control the number of colors in the colormap, use the **Colormap size** edit field on the **Postprocessing** page of the **Preference** dialog box (the default is 1024 colors).

The **Reverse** check box to the right of the colormap lists reverses the order of the colors in the selected colormap.

If your local COMSOL Multiphysics installation contains user-defined colormaps, these appear at the beginning of the **Colormap** list. The next section describes how you can create and install such colormaps.

CREATING AND INSTALLING CUSTOM COLORMAPS

You can define a custom colormap in a text file with three columns of RGB-values between 0 and 1. Each line containing a space-, comma-, or tab-separated list of three values defines a color in the colormap. Lines in the colormap file not defining a color are ignored, so you can use lines with comments and headers.

Place all colormap files in the directory `colormaps` under the preference directory (which by default is the subdirectory `.comsol/<version_tag>` under your home directory). The software then makes them available in the **Colormap** lists. The name appearing in the list is the name of the file except the file suffix (unless there is a conflict, in which case the whole file name is used). The list is updated each time you open the corresponding dialog box.

The script function `createcolormap` is a utility to create colormap files and save them to the `colormaps` folder. See the entry for `createcolormap` on page 47 of the *COMSOL Multiphysics Reference Guide*.

USING ROUGH PLOTS FOR LARGE MODELS AND QUICK VISUALIZATIONS

The “rough plot” feature makes it possible to do quick visualizations and plots of large 3D models. This feature generates plots with a bit lower quality than normal for faster and more memory-efficient rendering. To achieve this, COMSOL Multiphysics lowers the element refinement for postprocessing by 2 steps in 3D and by 1 step in 2D. It also turns off the calculation of smooth normals, which means that lighting does not affect the plot. One way of working with a large 3D model is to select the **Make rough plots** check box on the **General** page in the **Plot Parameters** dialog box while trying out different plots. When you get an interesting image, clear the check box to produce a high-quality plot. Use the preference setting **Make rough plots** on the **Postprocessing** page in the **Preferences** dialog box to control the default setting when creating new models.

Opening Models Faster Using Rough Plots

COMSOL Multiphysics does not store the current plot in the model but recreates it when you open the model again. Thus it can sometimes take a while to open a large 3D model that contains an advanced plot. To make COMSOL Multiphysics use rough plots when opening models, do the following:

- 1 Go to the **Options** menu and choose **Preferences**.
- 2 In the **Preferences** dialog box click the **Postprocessing** tab.
- 3 Select **Rough** in the **Plots when opening models** list.
- 4 Click **OK**.

It is also possible to avoid creating postprocessing plots when opening models. To do so, select **None** in the **Plots when opening models** list.

UPDATING A PLOT AND SAVING PLOT SETTINGS

- Click **OK** to apply the changes, plot the data, and close the dialog box.
- Click **Apply** to apply the changes and plot the data and keep the dialog box open.
- If you have not yet solved the model, clicking either **OK** or **Apply** saves the plot parameter settings for future plots.

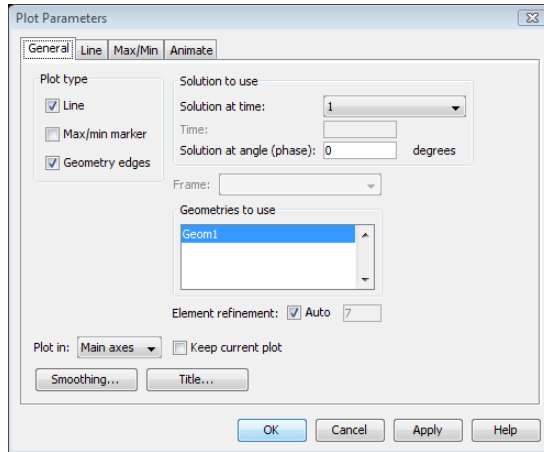


Figure 7-9: The 1D Plot Parameters dialog box.

Cross-Section Plots

Cross-section plots visualize a quantity as a family of plots on:

- Arbitrary planes (in 3D)
- An arbitrary line (in 2D or 3D)
- An arbitrary set of points (in 1D, 2D, or 3D)

The plots show the values over time, along a parametric solution, or for several eigenvalues.

To set common plot parameters, go to the **Cross-Section Plot Parameters** dialog box on the **General** page. For information on how to specify various cross-section plot types see “1D Cross-Section Plots” on page 498, “2D Cross-Section Plots” on page 526, and “3D Cross-Section Plots” on page 544.

To create a cross-section plot, go to the **Postprocessing** menu and choose **Cross-Section Plot Parameters**.

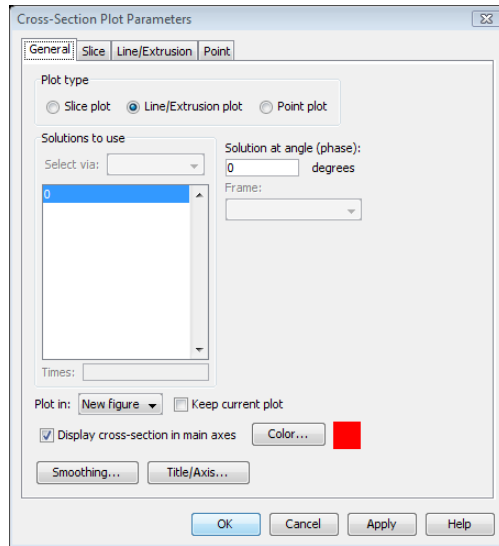


Figure 7-10: The 3D Cross-Section Plot Parameters dialog box.

QUICK CROSS-SECTION PLOTS

When working in 2D and 3D, you can obtain cross-section plots directly by defining points, lines, and faces using the mouse. To activate these quick plots, simply click the **Draw Points for Cross-Section Point Plot**, **Draw Line for Cross-Section Line Plot**, or **Draw Plane for Cross-Section Slice Plot** buttons on the Plot toolbar.

If the **Cross-Section Plot Parameters** dialog box is closed when you define a point, line, or face using the mouse, COMSOL Multiphysics launches a cross-section plot using the settings from the **Cross-Section Plot Parameters** dialog box; if the dialog box is open, the program updates the point, line, or slice position data.

To draw a new line or point, simply click the corresponding button again. To remove just the line or point, click the button twice.

Notice that for a 3D slice plot, the view angle might make it difficult to see the cross section. In that case, rotate the 3D plot slightly for a better view.

PLOTTING MULTIPLE DATA

Cross-section plots appear in separate figure windows. Select in which window to create the plot using the **Plot in** list; to visualize multiple data in the same figure, select the **Keep current plot** check box.

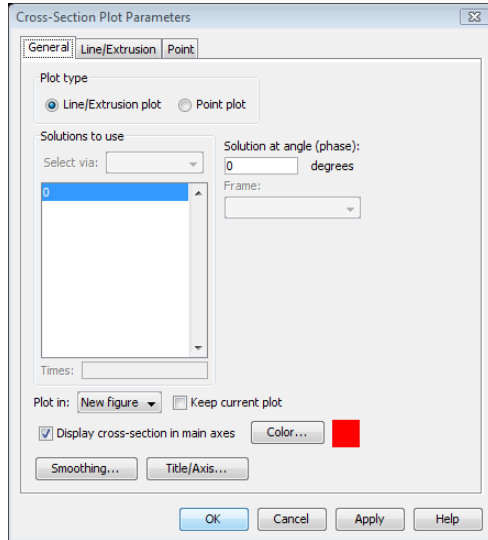


Figure 7-11: The 2D Cross-Section Plot Parameters dialog box.

To review other general plot settings, see “Visualization Using the Plot Parameters Dialog Box” on page 462.

Domain Plots

Domain plots visualize data on one or several geometric domains. You can plot for several eigenvalues, time steps or arbitrary times, or parameter values either as a family of line plots or by extruding them into a continuous plot.

To create a domain plot, go to the **Postprocessing** menu and choose **Domain Plot Parameters**.

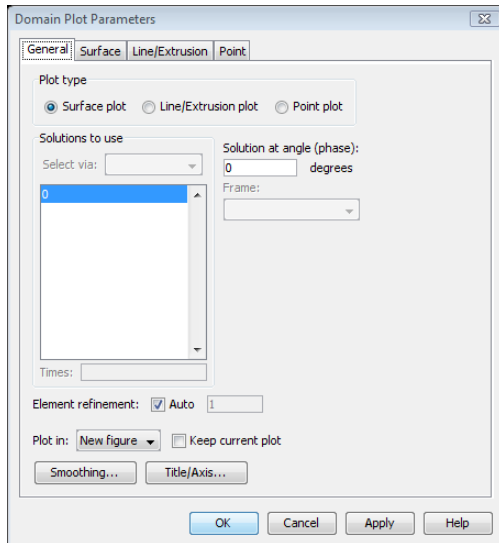


Figure 7-12: The 3D Domain Plot Parameters dialog box.

PLOTTING MULTIPLE DATA

Domain plots appear in separate figure windows. Select in which window the plot will appear using the **Plot in** list; to visualize multiple data in the same figure, select the **Keep current plot** check box.

To review other general plot settings, see “Visualization Using the Plot Parameters Dialog Box” on page 462.

Specifying Line and Surface Properties

When generating plots from the **Cross-Section Plot Parameters** and **Domain Plot Parameters** dialog boxes, you control the properties of line and extrusion plots using the **Line Settings** and **Surface Settings** dialog boxes, respectively.

LINE SETTINGS

COMSOL Multiphysics provides two places where you can specify line properties for cross-section plots and domain plots:

- Go to the **Cross-Section Plot Parameters** dialog box and then click the **Line/Extrusion** page
- Go to the **Domain Plot Parameters** dialog box and then click the **Point** page.

In either case, click the **Line Settings** button to open up the like-named dialog box, which contains the following settings:

- In the **Line color** list, select a fixed color or cycle through seven predefined colors: blue, green, red, turquoise, purple, yellow green, and black. To define a fixed color, click the **Color** button and make a selection in the **Line Color** dialog box.
- In the **Line Style** list, select a style or cycle through the available styles: solid, dotted, dashed, and dotted-dashed. Note that a dashed line can appear solid if the line resolution is too high; if so, try reducing the resolution (under **Cross-section line data** in the **Cross-Section Plot Parameters** dialog box).
- In the **Line Marker** list, select a marker or cycle through the available choices: triangles, squares, plus signs, asterisks, and stars. The default is no marker.
- Select the **Legend** check box if you want one for the plotted lines. The legend appears to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, the legend shows these values. For point plots, the legend displays the coordinate (or vertex number). For cross-section line plots with multiple parallel lines, each line gets a consecutive integer number that appears in the legend.

SURFACE SETTINGS

There are two places where you specify properties for a surface plot: go to either the **Domain Plot Parameters** or **Cross-Section Plot Parameters** dialog boxes, then proceed to the **Surface**, **Slice**, and **Line/Extrusion** pages. Now click the **Surface Settings** button.

Choosing Rendering and Fill Style

In the **Coloring and fill** area, first go to the **Coloring** list and choose between interpolated or flat rendering for the surface. Next, in the **Fill style** list choose between filled or wireframe rendering; the wireframe fill displays the surface as a triangular grid.

Note: The setting for fill style applies to the refined mesh elements for visualization, which can be much denser than the mesh the software uses for computation.

Colormaps and Color Scale Settings

Make a selection in the **Colormap** list. The default is the jet colormap.

Clearing the **Color scale** check box removes the color scale from the display.

Clicking the **Uniform color** button switches to a uniform color on the surface. To choose a color for the plot, click the **Color** button to open the **Surface Color** dialog box where you make a selection.

Creating Animations

You can animate any combination of standard COMSOL Multiphysics 2D and 3D plots in time or along a parametric solution.

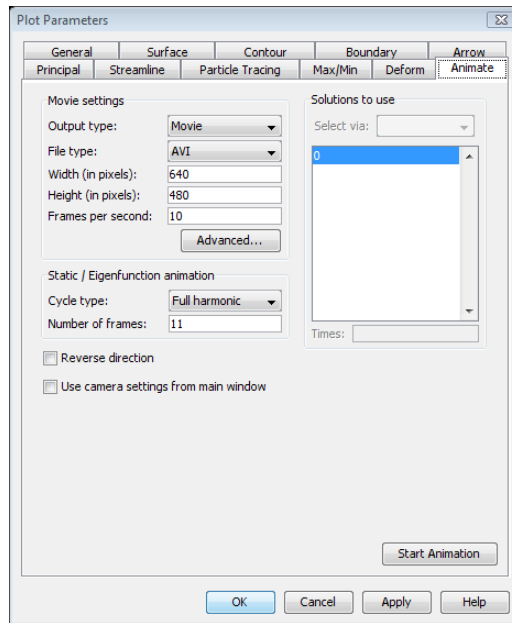


Figure 7-13: The animation settings in the Plot Parameters dialog box.

First go to **Movie settings** and select the output type. You can either create a movie or a sequence of image files, one per frame in the movie.

When creating an image sequence, select the file format of the image files in **File type**, and specify a file name in **File name**. You can also browse to a file or specify a new file name using the **Browse** button. The specified file name is the base for the file names of the created image files. For example, typing `image` in **File name** when **File type** is **PNG**, and the number of frames in the movie is 11, will create the files `image01.png`, `image02.png`, ..., `image11.png`. The names of the created files and their path appear in the message log.

When creating a movie, select the file format for the movie, either AVI, QuickTime, or GIF. The selection GIF creates an animated GIF file. Also indicate the movie's screen size in pixels and its frame rate. The screen size you enter will be increased if necessary to be at least 24-by-24 and a multiple of 8. Finally, select any subset of solutions to animate. For time-dependent problems, you can select **Interpolated times** from the **Select via** list, and enter any vector of times in the **Times** edit field, such as `linspace(0, 1, 100)` or `0:0.02:1` in the **Times** edit field.

To quickly get an animation of the current plot, click the **Start Animation** button.

When animating static and eigenvalue solutions, you must specify the animation cycle in the **Cycle** list. The available choices are:

- **Full harmonic** (a full sine wave)
- **Half harmonic** (half a sine wave)
- **Linear** (a linear ramp)

This provides a way of visualizing the time evolution of a time-harmonic model using a full sine wave as the animation cycle.

In the **Number of frames** edit field you specify how many frames the animation should include. The display cycles 360 degrees starting from the angle you specify in the **Solution at angle (phase)** edit field on the **General** page.

It is also possible to rewind the movie by selecting the **Reverse direction** check box.

The **Use camera settings from main window** check box controls the camera behavior when animating 2D height plots and is only visible in 2D mode. The default behavior of the camera is to zoom out to fit all data of the animation in the window. Select this option to override this behavior and use the camera settings from the main window instead.

When using the MATLAB interface, the **Make movie in MATLAB** check box is available. If you select it before you click **Start Animation**, the animation appears in a figure window in MATLAB, similar to an animation using the `postmovie` command.

ADVANCED ANIMATION SETTINGS

For the AVI and QuickTime file formats, you can choose which encoding format to use to encode the video in the **Advanced Animation Settings** dialog box opened by clicking the **Advanced** button.

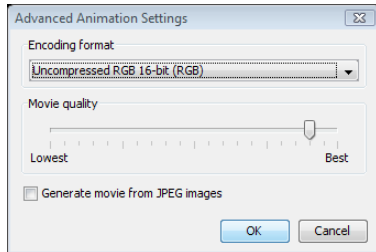


Figure 7-14: The Advanced Animation Settings dialog box.

Both AVI and QuickTime are file formats that can contain video encoded in different ways. The choice of codec (COder/DECoder) determines the quality and file size of the movie. You can also adjust the quality using the **Movie quality** slider.

The **Generate movie from JPEG images** option tells COMSOL Multiphysics to make a JPEG image for each frame and then use these to create the movie. This can work in some cases where the normal method fails. Select this option if you experience trouble with the movie generation.

The **Advanced Animation Settings** dialog box displays a list of detected compatible codecs. Choose the codec to use to encode the video from this list. Which codecs that are available depends on your operating system, if AVI or QuickTime is chosen as output file type, and if the **Generate movie from JPEG images** option is selected or not. On Microsoft Windows operating systems, installed Video Compression Manager (VCM) codecs might show in the list, marked with the prefix “External:”.

Table 7-4 compares the most commonly available codecs and in which contexts they are available.

TABLE 7-4: COMMONLY AVAILABLE CODECS

ENCODING FORMAT	AVI	QUICKTIME	PLAY IN COMSOL MOVIE PLAYER	PLAY IN WINDOWS MEDIA PLAYER	PLAY IN QUICKTIME PLAYER
Uncompressed RGB	√	√	√	√	√
Cinepak	√	√	√	√	√
Intel Indeo 3.2	√	√	√	√	√
Uncompressed YUV	√		√	√	√
IYUV	√		√	√	√
JPEG		√	√		√
H.263	√	√	√		√
Motion JPEG	√	√	√		

Note: Not all of these codecs might appear on your system. Compression is only available on 32-bit Windows, 32-bit Linux, and 32-bit Solaris.

Plotting Values While Solving

This section describes probe plots that describe scalar quantities while solving. You can also make standard plots of field variables while solving time-dependent, parametric, and eigenvalues problems. See “Plotting the Solution While Solving” on page 424 for more information about this option.

If you use the time-dependent or parametric solver (specified in the **Solver Parameters** dialog box), you can plot the value of certain quantities versus time or the parameter value while the solver is running. These quantities include

- The value of an expression in a point (probe) in the a subdomain in the model, specified by its coordinates.
- The value of an expression in a point (probe) that is a point (vertex) in the geometry.

- The value of the integral of an expression over arbitrary subdomains, boundaries, or (in 3D) edges.
- The value of a globally defined variable, such as a solution to an ODE or a coupling variable with global destination.

These probe plots appear on the **Plot** page of the **Progress** dialog box, which COMSOL Multiphysics displays while the solver is running. You can specify several plots of these types, in which case each plot can appear in its own coordinate system or be displayed in the same coordinate system. You specify this using the **Plot all plots in the same axis** check box in the **Probe Plot Parameters** dialog box.

While the solver is running, you can copy the plots to a separate figure window by clicking the appropriate **Copy to Figure** button. When the solver has finished, all the plots are displayed in a separate figure window automatically. To turn off this behavior, clear the **Copy probe plots to separate figure window when solver has finished** check box in the **Preferences** dialog box.

The axis limits in the coordinate systems are dynamic and can thus change while the solver is running. To make probe plots with fixed axis limits, use the **Manual specification of axis limits** setting, present in the plot definition area for each of the types of probe plot.

The following figure shows an example of how a probe for a velocity component appears in the **Progress** dialog box.

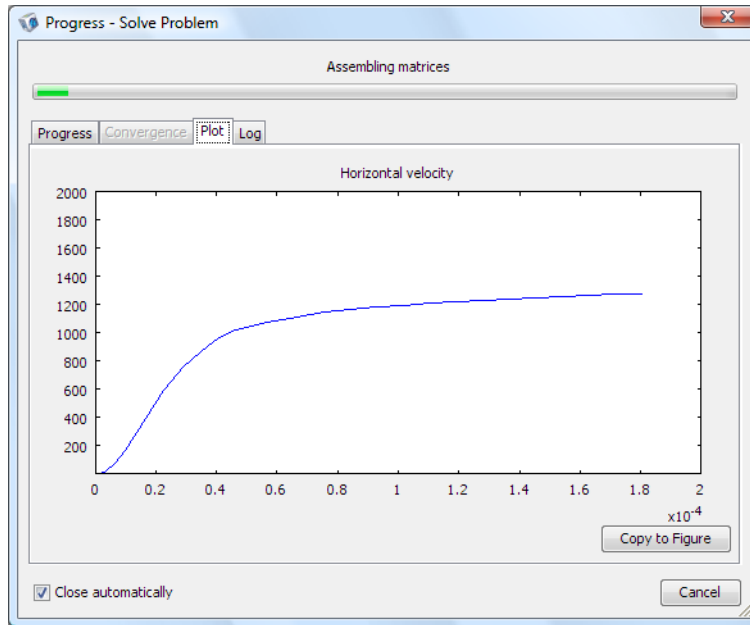
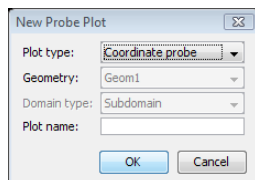


Figure 7-15: The Progress dialog box showing a probe plot.

For the **Coordinate Probe Plot** you can use the **Draw Points for Coordinate Probe Plot** toolbar button, available in the Subdomain, Boundary, Edge, and Point selection modes, to quickly specify the points in which to plot. Click this toolbar button, and then use the mouse to click where you want the points to be positioned. When finished, deselect the toolbar button by clicking it again. This creates a probe plot of the default subdomain plot quantity.

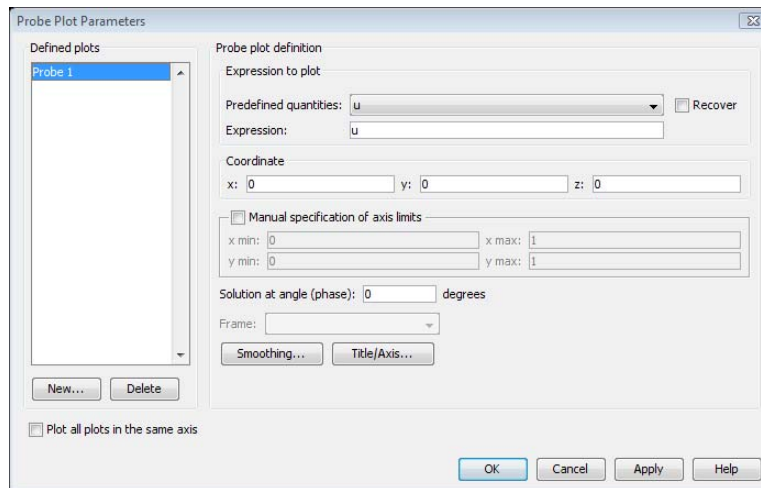
For all plot types, use the following steps to specify which quantities to plot:

- 1 Select the menu item **Postprocessing>Probe Plot Parameters**, and click **New**. This opens the **New Probe Plot** dialog box.



- 2 From the **Plot type** list, select which type of plot to define.
 - Select **Coordinate probe** to plot the value of an expression in a point (probe) in the a subdomain in the model, specified by its coordinates.
 - Select **Point probe** to plot the value of an expression in a point (probe) that is a point (vertex) in the geometry.
 - Select **Integration** to plot the value of the integral of an expression over arbitrary subdomains, boundaries, or (in 3D) edges.
 - Select **Global** to plot the value of a globally defined variable, such as a solution to an ODE or a coupling variable with global destination.
- 3 For **Coordinate probe** plots and **Integration** plots, specify in which geometry the plot is defined (for multigeometry models). For **Integration** plots, specify on which geometry domain to integrate in the **Domain type** list. Optionally, give the plot a name. An automatic name is generated if the **Plot name** field is left blank. This name appears in the list of defined plots in the **Probe Plot Parameters** dialog box.
- 4 The plots appear in the **Defined plots** list in the **Probe Plot Parameters** dialog box. For each defined plot, specify the plot options in the right area of the **Probe Plot Parameters** dialog box. See the following sections for the plot options for each plot type.

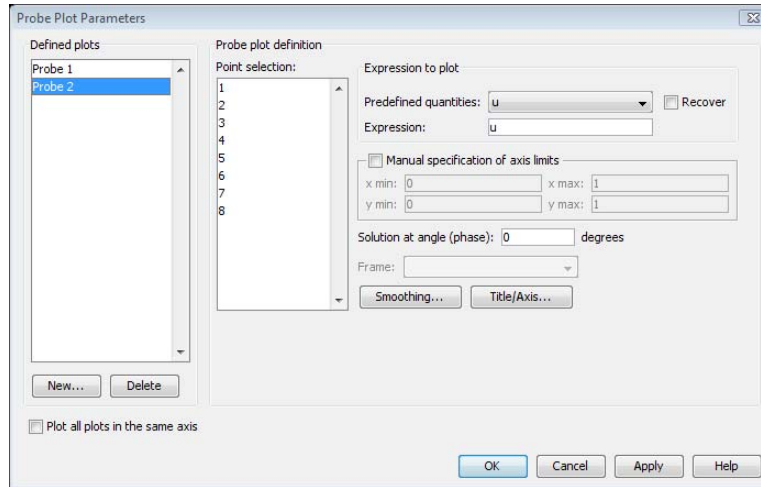
COORDINATE PROBE PLOTS



The settings in the **Probe plot definition** area are analogous to the settings on the **Point** and **General** pages of the **Cross-Section Plot Parameters** dialog box (see “Cross-Section

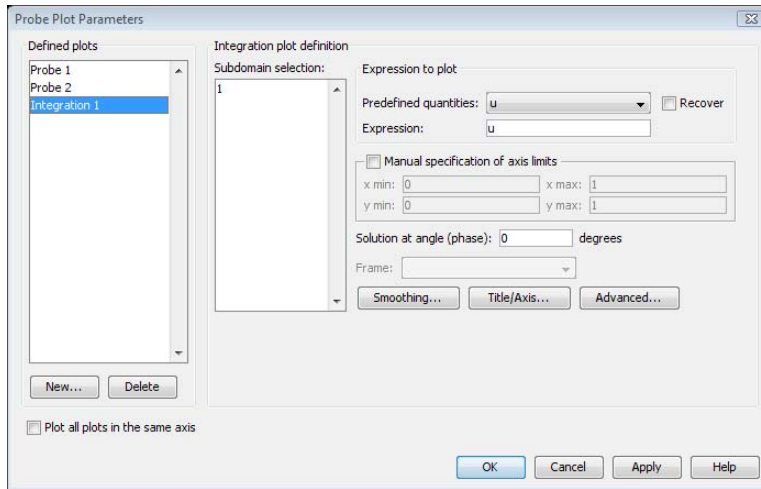
Plots” on page 473). Note that only a single coordinate is specified here. To plot in several points, define a new coordinate probe plot for each point.

POINT PROBE PLOTS



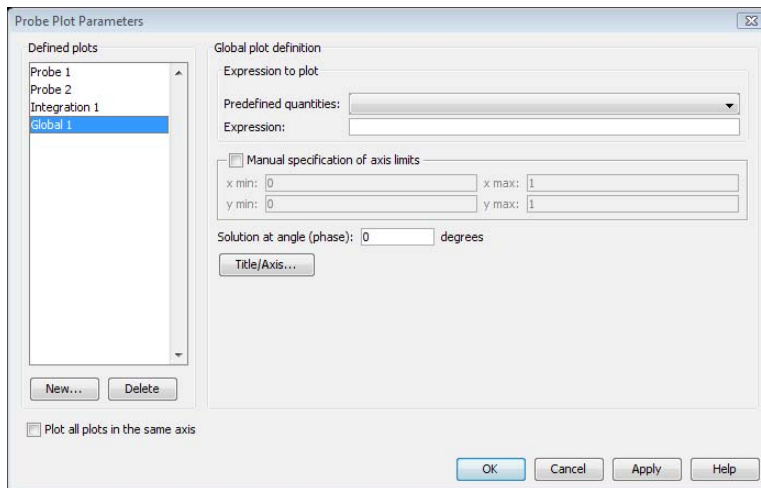
The settings in the **Probe plot definition** area are analogous to the settings on the **Point** and **General** pages of the **Domain Plot Parameters** dialog box (see “Domain Plots” on page 475). Note that only a single geometry point is specified here. To plot in several points, define a new point probe plot for each point.

INTEGRATION PLOTS



The settings in the **Integration plot definition** area are analogous to the settings in the integration dialog boxes (see “Computing Quantities Using Integration” on page 456).

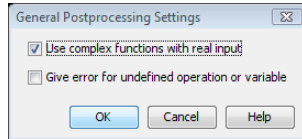
GLOBAL PLOTS



The settings in the **Global plot definition** area are analogous to the settings in the **Global Variables Plot** dialog box (see “Plotting Global Expressions” on page 461).

General Postprocessing Settings

The **General Postprocessing Settings** dialog box contains settings that apply to all postprocessing: all plot types, integration, data display, and export of postprocessing data.



UNDEFINED OPERATIONS OR VARIABLES

By default, COMSOL Multiphysics plots an expression wherever it is defined. If an expression contains variables that are not defined everywhere, the plot only covers the areas where the expression is defined. When the option **Give error for undefined operation or variable** is selected, COMSOL Multiphysics instead gives an error message for partially defined expressions containing information about variables that are not defined everywhere. If you select this option, you also get an error for undefined operations such as unexpected complex output from a function with real input as described below.

Thus, if a plot is not rendered in all areas where you expected a plot, use these settings to get information about which operation or variable that caused the empty areas.

COMPLEX-VALUED FUNCTIONS

If you allow complex outputs from real inputs in elementary functions such as `sqrt`, `log`, and `asin` during postprocessing, then select the **Use complex functions with real input** check box. In this case, all plot types show the real part of complex-valued quantities. When this option is not used, these functions (for example, `sqrt(-1)`) return the value NaN instead of the complex value. For plotting, this generates a partial plot with no rendering in the areas with NaN values. If you select the **Give error for undefined operation or variable** check box and clear the **Use complex functions with real input** check box, COMSOL Multiphysics gives an error message if a real input to a function generates a complex-valued output.

Exporting Postprocessing Data to a File

As a complement to the visualizations, you can export results data to a text file for further postprocessing.

To export the data to a text file, follow these steps:

- 1 Select the menu item **File>Export>Postprocessing Data**.

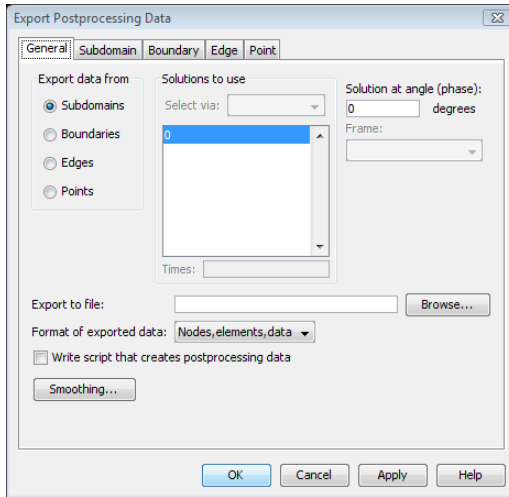


Figure 7-16: The *Export Postprocessing Data* dialog box.

- 2 In the **Export Postprocessing Data** dialog box, enter the name of the file in the **Export to file** edit field or click the **Browse** button to use the **Save Data As** dialog box.
- 3 In the **Export data from** area, select the domain type for the export: **Subdomains**, **Boundaries**, **Edges** (3D), or **Points** (2D).
- 4 If applicable, select the solutions to use (multiple solutions are permissible) when exporting the data to the file.
- 5 COMSOL Multiphysics can export the data in several possible formats, which you select from the **Format of exported data** list. You have the following options:
 - **Nodes, elements, data** (available for all domain types): This option writes the coordinates for the nodes (in the % `Coordinates` section, where the i th row contains the coordinates for node i); how the nodes are connected (in the % `Elements` section, where the j th row contains the indices i_1, \dots, i_n (n is the number of nodes per element) of the nodes that belong to element j); and the

data values at each node point (in the % **Data** section, where the i th row contains the data values at node i).

If you select more than one solution in the **Solutions to use** list, each solution appears in a separate column in the % **Data** section, with an extra header line showing the associated parameter values.

- **Grid, data** (available only for subdomains). The first lines in the data file are the grid levels in each direction. The software then writes the data as a matrix in a format similar to that produced by the `meshgrid` function in COMSOL Script and MATLAB:

```
% Grid
x grid points separated by spaces
y grid points separated by spaces (optional)
z grid points separated by spaces (optional)
% Data
Data values separated by spaces
```

Each row contains values for different x grid points for fixed y and z . The rows first increase the y grid value and then the z grid value.

- **Coordinates, data** (available for all domain types). Using this option, the software writes an initial header line and then the node point coordinates (in 1, 2, or 3 columns) followed by the data values in separate columns (one for each solution).
- 6 Optionally, to write an M-file with commands that create a structure of postprocessing data, select the **Write script that creates postprocessing data** check box. For a description of the structure's format, see the entry for `posteval` on page 380 of the *COMSOL Multiphysics Reference Guide*.

7 Click the **Subdomain** tab to access the subdomain data export page:

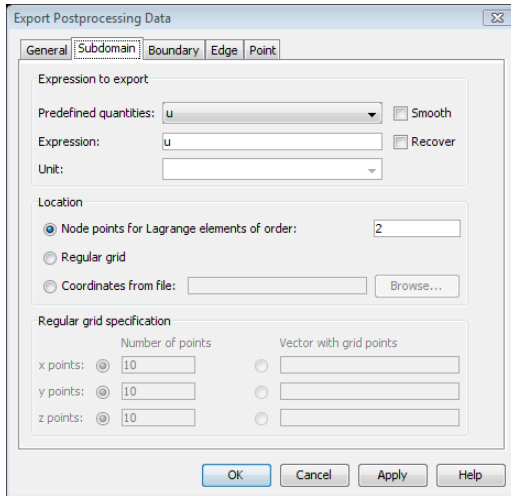


Figure 7-17: The subdomain settings in the Export Data dialog box.

- 8 Enter the name of the postprocessing expression that specifies the data on subdomains that you want to save to a file.

You can choose to evaluate the expression at one of the following locations:

- Node points for Lagrange elements of a certain order.
- On a regular grid specified as the number of grid lines in each direction or as vectors with specific values of the grid points' coordinates.
- Coordinates read from an input file with the following format: one coordinate per row with each component separated by spaces.

For dimensions lower than the subdomains (meaning edges, boundaries, points, and faces), only the first item is available. Evaluation at a point itself is available only at the points in the geometry.

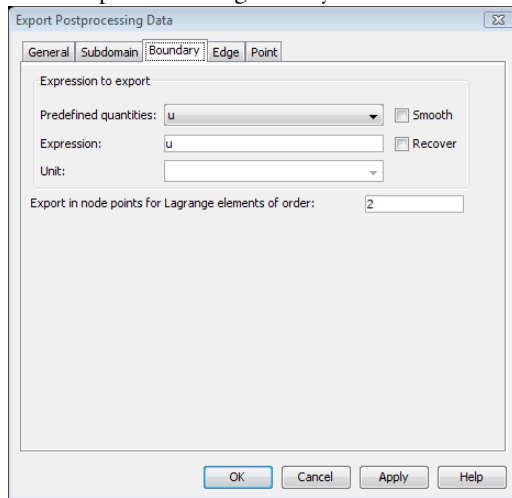


Figure 7-18: The boundary settings in the Export Data dialog box.

Exporting the Current Plot to a File

To export the current plot data in the main axes, select the menu item **File>Export>Current Plot**. This opens the **Save Current Plot in ASCII file** dialog box.

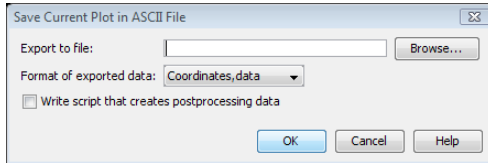


Figure 7-19: The plot export dialog box.

The data format is either “Nodes, elements, data” or “Coordinates, data”, specified in the **Format of exported data** list (see “Nodes, elements, data” on page 488 and “Coordinates, data” on page 489). Here, you can also select **Write script that creates postprocessing data** (described in “Exporting Postprocessing Data to a File” on page 487). This way you can generate an M-file that creates a structure of postprocessing data representing the data in the current plot. A description of the structure’s format appears under the entry for `posteval` on page 380 of the *COMSOL Multiphysics Reference Guide*.

If the current plot contains several components, the corresponding postprocessing data is written to the file component by component. Note the following special cases:

- Plot components with uniform color have no associated data section, with one exception: for contour plots with uniform color, the data section contains the isolevel values.
- The data for 2D height data plots appear in the third coordinates column.
- For arrow plots, the `% Coordinates` section is followed by a `% Vector` data section of the same length whose columns contain the component values of the vector-valued quantity.

Exporting Data to COMSOL Script or MATLAB

It might be desirable to export a model to COMSOL Script or MATLAB for many reasons, for instance, if you want to obtain the values of the solution at several points. To export an entire model as an *FEM structure*:

- 1 Select the menu item **File>Export>FEM Structure**.

- 2 From the command line, use the functions `posteval` and `postinterp` to compute numerical values for any expression, or use `postint` to perform integration.

See the entries on `posteval`, `postinterp`, and `postint` in the “Command Reference” chapter of the *COMSOL Multiphysics Reference Guide* for more information about postprocessing commands.

Exporting Images

To export the current plot as an image select the menu item **File>Export>Image**. Using the **Export Image** dialog box, you can control font sizes, line widths, and what to include in the image you export. To speed up the image-generation process, a preview feature and image rendering information are available.

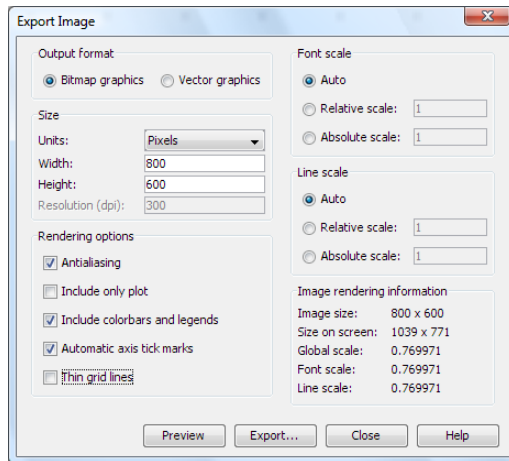


Figure 7-20: The Export Image dialog box.

SETTINGS IN THE EXPORT IMAGE DIALOG BOX

Output Format

In the **Output format** area, click the **Bitmap graphics** button to export an image using a bitmap-based formats such as TIFF and JPEG. You select the format after clicking the **Export** button. Click the **Vector graphics** button to save the image as an EPS file (Encapsulated PostScript).

Image Size

In the **Image size** area, you can specify the size of the image. Select the unit from the **Units** list: inches, centimeters, or pixels (pixels are available for bitmap graphics only).

When using inches or centimeters, you can set the size using the **Width**, **Height**, and **Resolution (dpi)** edit fields. When you use pixels as the unit, the **Resolution (dpi)** edit field is not available

Scaling

The settings in the **Font scale** and **Line scale** areas affect the scaling between the plot's size on the screen and the size in the image (size = number of pixels):

- Click **Auto** to use the global scale (you see its value in the **Image rendering information** area; also see “The Global Scale” on page 495) if you specify the size in pixels (the software scales text, lines, and other graphics equally). If you specify the size in centimeters or inches, the automatic scale is based on the resolution that you. The font size and line width that you specify when creating the plot should be preserved if you export an image using a certain resolution in dpi (dots per inch) and import it to a document as an image using the same dpi number (a text with a certain size in the plot will look like a text with the same size in the document).
- Click **Relative scale** to use a total font scale that is the automatic scale times the relative scale that you specify.
- Click **Absolute scale** to use a total font scale that is equal to the absolute scale that you specify.

RENDERING OPTIONS

The following settings are available in the **Rendering options** area:

Antialiasing

Select the **Antialiasing** check box to reduces stairstep-like lines (jaggies) and makes lines and edges look smooth.

Include Only Plot

Select the **Include Only Plot** check box to include only the graphics objects in the drawing area, excluding colorbars, axes, tick marks, titles, and labels.

Include Colorbars and Legends

Select the **Include Colorbars and Legends** check box to include colorbars and legends, if present.

Automatic Axis Tick Marks

Select the **Automatic axis tick marks** check box to hide axis tick marks if they overlap. Clear this check box if you want make sure that the image has the same axis tick marks as you see on the screen.

Rendering Thin Grid Lines

Select the **Thin grid lines** check box to render thin grid lines compared to other lines in the image. Use this option if you think that the grid is too dominating in the image.

Including the 3D Axis

Select the **Include 3D axis** check box to include the grid and coordinate system in the image.

THE GLOBAL SCALE

The global scale, which you find in the **Image rendering information** area, is the scale between the size of the plot on screen and the size of the image (size = number of pixels).

Using Max/Min Markers

Maximum/minimum markers are available for standard plots in all space dimensions.

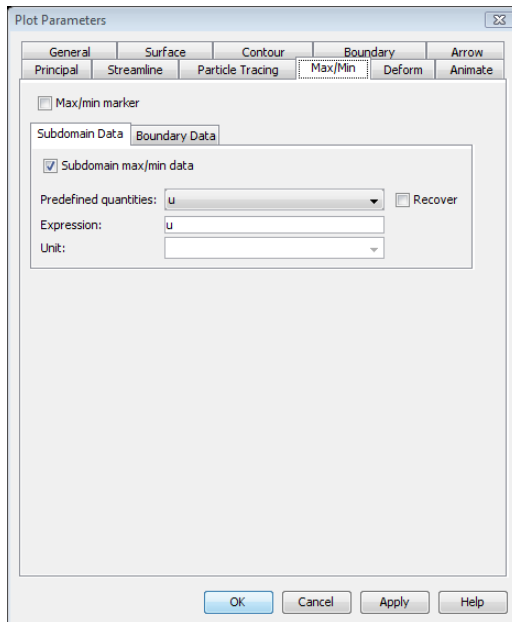


Figure 7-21: The maximum/minimum marker settings in the Plot Parameters dialog box.

You can set these markers for arbitrary expressions independently for each space dimension.

Postprocessing 2D Axisymmetric Models in 3D

To postprocess the solution to a 2D axisymmetric model in full 3D, first create the corresponding 3D geometry by revolving the axisymmetric geometry about the axis of symmetry. Then use extrusion coupling variables to map the solution components from the rz -plane to the 3D coordinates. See “Brief Examples of Extrusion Coupling variables” on page 283 for information about the coordinate transformation and “2D Axisymmetric Transient Heat Transfer” on page 196 in the *COMSOL Multiphysics Modeling Guide* for an example of how to postprocess an axisymmetric model in 3D. The transformation of a vector-valued solution such as velocity can also involve some trigonometry and is less straightforward than the transformation for scalar quantities.

Postprocessing Results in 1D

1D Postprocessing Overview

Line plots visualize a 1D model quantity as an xy -plot. The x coordinate corresponds to a 1D geometric coordinate, and the y coordinate corresponds to the quantity's numeric value. COMSOL Multiphysics can animate standard plots in time or along a parametric solution.

- Use cross-section plots in 1D to visualize the solution in one or several points in time or along a parametric solution.
- Use domain plots to visualize data on one or several geometric domains in time, along a parametric solution as a family of plots, or by extrusion to a continuous plot.

1D Standard Plots

The xy line plot is the standard type in 1D. The x coordinate corresponds to the 1D geometric coordinate, and the y -coordinate corresponds to the visualized quantity.

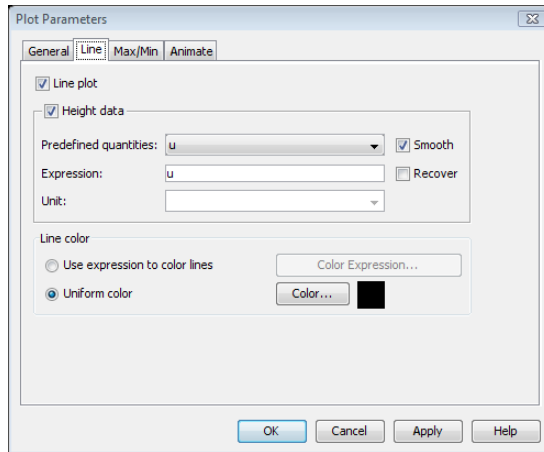


Figure 7-22: The line plot settings in the 1D Plot Parameters dialog box.

Select the **Line plot** check box to enable this plot type. The **Height data** setting determines the quantity on the y -axis. Either select from application mode dependent quantities or enter an expression containing variables. Select the **Smooth** check box to enforce continuity on discontinuous data.

The line plot's color is black by default. In the **Line color** area you can change it to a different uniform color or have the software change the plot's color based on an expression.

1D Cross-Section Plots

ID POINT CROSS-SECTION PLOTS

A 1D cross-section point plot visualizes a quantity in one or several points in time, along a parameter range, or for several different eigenvalues.

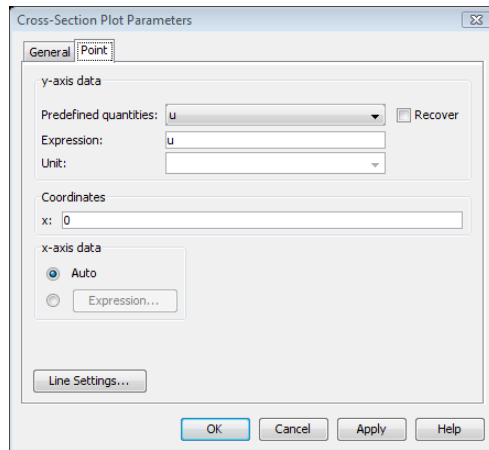


Figure 7-23: The cross-section plot parameters in the 1D Cross-Section Plot Parameters dialog box.

Determine the quantity to plot using settings in the **y-axis data** area; either select from predefined quantities or enter an expression containing variables.

Control the quantity being plotted on the x -axis in the **x-axis data** area. If **Auto** is selected, the x -axis corresponds to time, parameter values, or the eigenvalue number. To set up an arbitrary subdomain-defined quantity on the x -axis, click the **Expression** button. In the **Coordinates** area go to the **x** edit field and supply x -coordinate values for the plot.

You can enter one or several space-separated values or a vector of coordinates, for example, 0:10:100.

Use the **Line Settings** dialog box to adjust the plot's color, style, and markers.

1D LINE/EXTRUSION DOMAIN PLOTS

This plot type visualizes the solution in one or several subdomains.

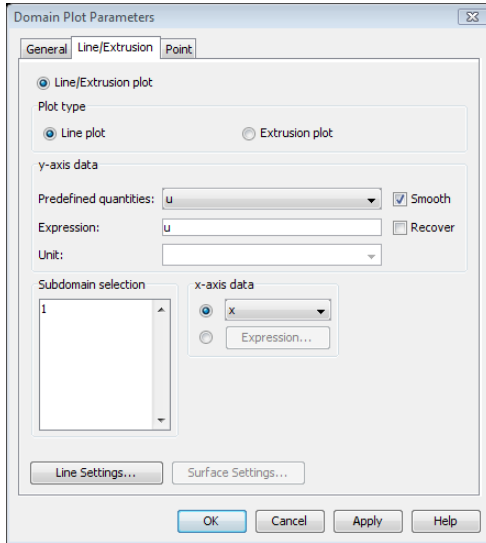


Figure 7-24: The line and extrusion plot parameters in the 1D Domain Plot Parameters dialog box.

To create a line plot or extruded surface plot on a 1D domain, do the following steps:

- 1 In the **Plot type** area click the **Line plot** check box to create a superimposed line plot, or click the **Extrusion plot** check box to have the software extrude the line plot into a continuous surface plot.
- 2 In the **y-axis data** area choose the quantity to display. Either select from predefined quantities or enter an expression containing variables.
- 3 Select the domains for the plot in the **Subdomain selection** list.
- 4 The software provides two ways to determine values on the x -axis. In the **x-axis data** area, either click the top button and select **x** (the values on the x -axis in the model geometry) or **Arc-length** (the arc length starts at 0 and represents the length of the arc); or click the bottom button, then click **Expression** to set x -axis data using another quantity.
- 5 For y -axis data, select a predefined quantity or enter an expression.

The **Line Settings** dialog box allows adjustment of the plot's color, style, and markers, while the **Surface Settings** dialog box allows adjustment of the surface on an extrusion plot.

1D POINT DOMAIN PLOTS

This plot type visualizes the solution in one or several geometric points or vertices.

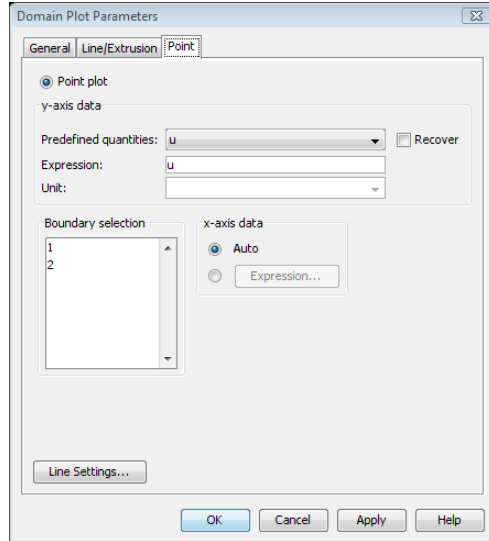


Figure 7-25: The point plot settings in the 1D Domain Plot Parameters dialog box.

To create a 1D domain plot for points, do the following step:

- 1 In the **y-axis data** area choose the quantity to display. Either select from predefined quantities or enter an expression containing variables.
- 2 Select the points to plot in the **Boundary selection** list.

The **Line Settings** dialog box allows adjustment of the plot's color, style, and markers.

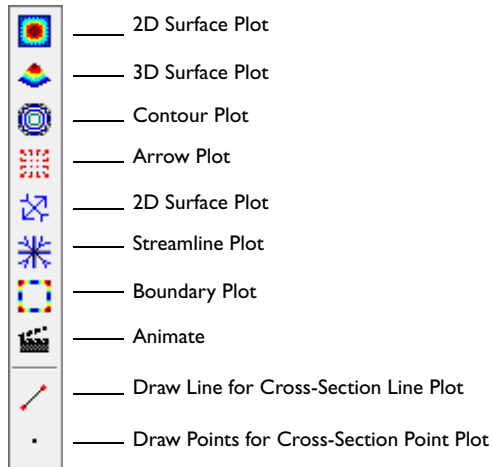
Postprocessing Results in 2D

2D Postprocessing Overview

Standard plots in 2D visualize the solution as part of the geometry. It can appear as a combination of 2D or 3D surface plots, a contour plot, an arrow plot, a deformed shape plot, a streamline plot, a particle tracing plot, or a line plot. COMSOL Multiphysics can animate these standard plots in time or along a parametric solution. Furthermore, cross-section plots can visualize a quantity as a family of plots in an arbitrary line or in an arbitrary set of points in time or along a parametric solution. Domain plots visualize quantities on one or several geometric domains in time or along a parametric solution as a family of plots.

Using the 2D Plot Toolbar

The following plot shows the Plot toolbar for 2D models:



Clicking any of the top seven buttons generates a corresponding plot using the current settings for that plot type in the **Plot Parameters** dialog box. Click the **Animate** button to record and play a movie. Use the **Draw Line for Cross-Section Line Plot** and **Draw Points for Cross-Section Point Plot** buttons to interactively create cross-section line and point plots. For more information, see “Quick 2D Cross-Section Line and Point Plots” on page 528.

2D SURFACE PLOTS

The surface plot is the default in 2D. It displays a quantity as a colored 2D and optionally as a 3D surface.

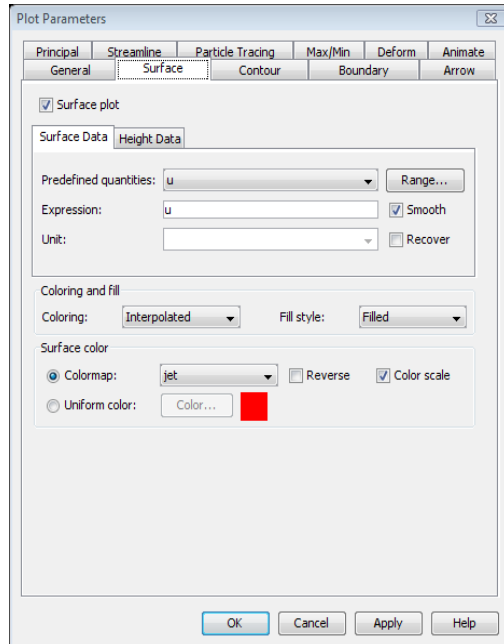


Figure 7-26: The surface plot settings in the 2D Plot Parameters dialog box.

To create a surface plot:

- 1 Select the **Surface plot** check box to enable this plot type.
- 2 In the **Surface data** area you define the quantity or expression to visualize using a colored surface. Either select from predefined quantities or enter an expression containing variables.
- 3 It is also possible to extend a surface plot to a 3D plot by setting the *z*-axis quantity. To do so, select the **Height data** check box. Define the quantity by either selecting from predefined quantities or entering an expression containing variables.

Quick 2D Surface and Height Plots

Click the **2D Surface Plot** button on the Plot toolbar to get a quick plot of this type. This action disables all other plot types and clears the **Height data** check box. The plot takes its other settings from the **Surface** page.

Click the **3D Surface Plot** button on the Plot toolbar to get a quick plot of this type. This action disables all other plot types and selects the **Height data** check box.

Another way to get quick plots is to go to the **Postprocessing** menu, point to **Quick Plots**, and then click **2D Surface Plot** or **3D Surface Plot**.

Surface Plot Properties

To learn about the settings in the **Coloring and fill** and **Surface color** areas, see “Surface Settings” on page 477.

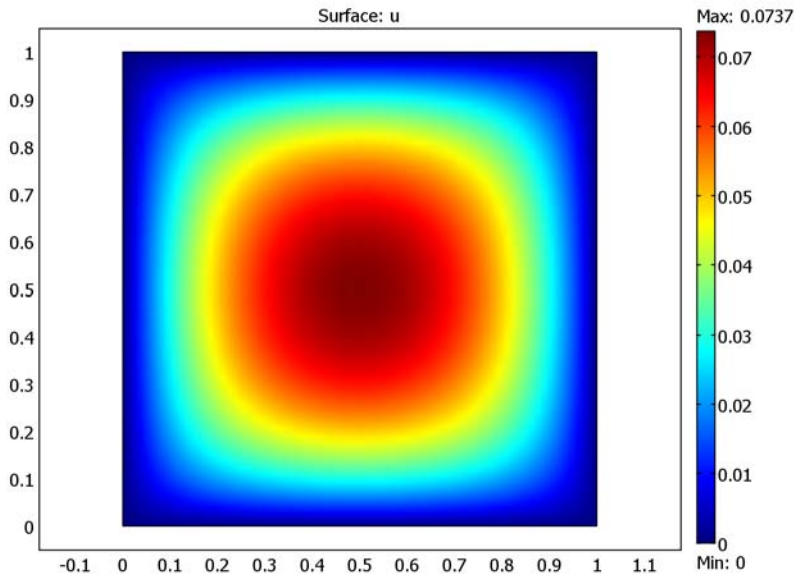


Figure 7-27: A surface plot of the solution to a 2D Poisson’s equation.

2D CONTOUR PLOTS

A contour plot displays a quantity as a set of colored contour lines. The selected quantity has a constant value on these contour lines, optionally with a 3D height. The software can also color the contour lines based on an independent quantity. There is also an option to fill the area between the contour lines with a constant color.

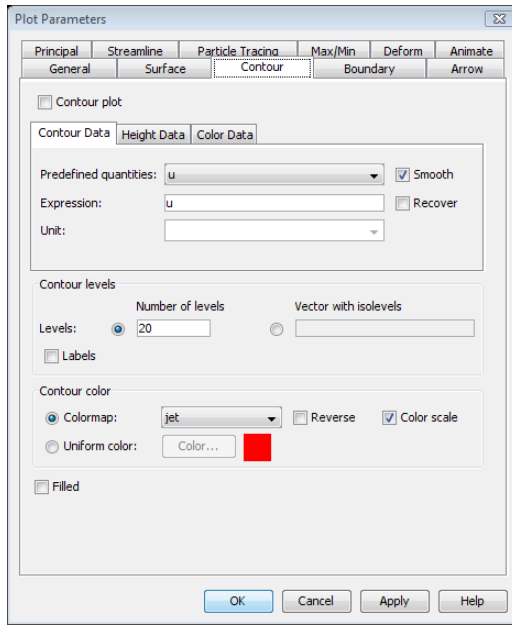


Figure 7-28: The contour plot settings in the 2D Plot Parameters dialog box.

To create a contour plot:

- 1 Select the **Contour plot** check box to enable this plot type.
- 2 Define the data to visualize in the **Contour data** area. Either select from predefined quantities or enter an expression containing variables. The quantity has a constant value on the resulting contour lines.
- 3 Select the **Smooth** check box to enforce continuity on discontinuous data.
- 4 To extend the surface plot to a 3D plot using a z -axis quantity, select the **Height data** check box. Once again, either select from predefined quantities or enter an expression containing variables.

Quick Contour Plots

Click the **Contour Plot** button on the Plot toolbar to quickly bring up that type of plot; this action disables all other plot types. Configure the plot using settings on the **Contour** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Contour Plot**.

Filled Contour Plots

To create a filled contour plot, select the **Filled** check box. This creates a plot with as many bands of constant color as you have specified in the **Number of levels** edit field. Alternatively, the filled contour plot is a contour plot with the levels specified in **Vector with isolevels** edit field, where COMSOL Multiphysics then fills the area between the contour lines. The color of each filled area represents the data interval between the contour lines. The color scale shows these data intervals.

Defining Contour Levels and Adding Labels

Go to the **Contour levels** area to adjust the like-named settings. Click the button to the left of the **Number of levels** edit field to supply an integer that indicates the number of levels (the default is 20). Alternatively, click the button to the left of the **Vector with isolevels** edit field to enter a vector, for example, 0:10:100. To include contour labels, select the **Labels** check box in the **Contour levels** area.

Coloring Contours

By default, COMSOL Multiphysics colors contour lines based on settings in the **Contour data** area, and it makes each line a uniform color. To instead color the contour lines based on an independent quantity, select the **Color data** check box and enter a predefined quantity or an expression.

For most plot types, the default for color mapping is to spread the range of colors across the entire range of variable values. Thus, in any given limited range of values it might be difficult to discern any differences. To overcome this effect, you can set the display to have the full range of colors map onto a subset of the variable values. To do so, click the **Range** button and in the resulting dialog box set the color range to zoom in on specific contours to reveal small differences at an interesting location.

Colormaps and Color Scale Settings

Make a selection in the **Colormap** list. The default colormap is **hsv**.

Clearing the **Color scale** check box removes the color scale from the display.

Clicking the **Uniform color** button switches the contours to a uniform color, whereas clicking the **Color** button opens the **Contour Color** dialog box where you can select a color for the plot.

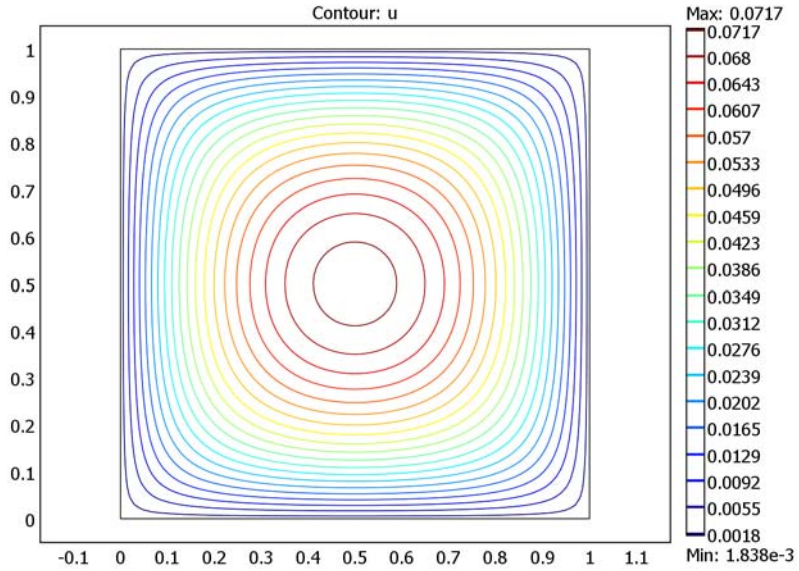


Figure 7-29: A contour plot of the solution to a 2D Poisson's equation.

2D BOUNDARY PLOTS

A 2D boundary plot visualizes a quantity on the boundary of the domain or on any 1D domain within a 2D geometry.

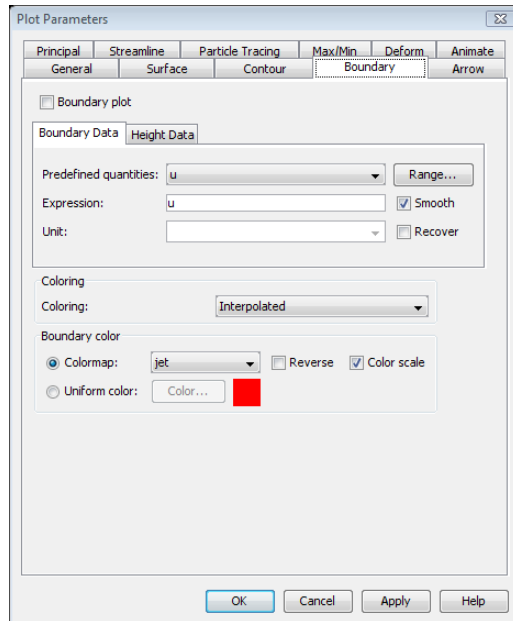


Figure 7-30: The boundary plot settings in the 2D Plot Parameters dialog box.

To create a boundary plot:

- 1 Select the **Boundary plot** check box to enable this plot type.
- 2 In the **Boundary data** area you define the boundary color. Either select from predefined quantities or enter an expression containing variables.
- 3 To extend the boundary plot to a 3D plot, select the **Height data** check box and choose the z-axis quantity. Either select from predefined quantities or enter an expression containing variables.

Quick Boundary Plots

Click the **Boundary Plot** button on the Plot toolbar to quickly generate that type of plot; this choice disables all other plot types. The plot takes its settings from the **Boundary** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Boundary Plot**.

Coloring Settings

To choose between interpolated and flat rendering of the boundaries, go to the **Coloring** list. Notice that this setting applies to the refined mesh elements for visualization, which can be much denser than the mesh used for computation.

Select the colormap to use in the **Colormap** list; the default is the **jet** colormap.

Clearing the **Color scale** check box removes the color scale from the display.

Clicking the **Uniform color** button switches to a uniform color on the boundaries, whereas clicking on the **Color** button opens the **Boundary Color** dialog box where you select a color for the plot.

2D ARROW PLOTS

A 2D arrow plot visualizes a 2-component vector quantity on the domain.

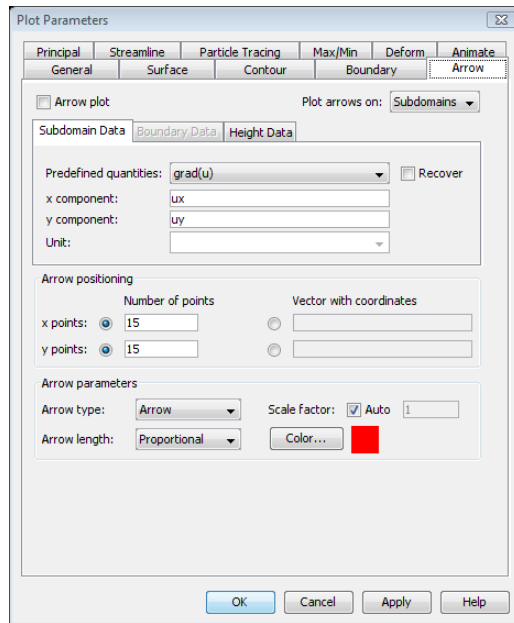


Figure 7-31: The arrow plot settings in the 2D Plot Parameters dialog box.

To create an arrow plot, do the following steps:

- 1 Select the **Arrow plot** check box to enable this plot type.

- 2 In the **Arrow data** area, indicate which data defines arrow lengths and directions. Either select from predefined vector fields, or enter a variable or an expression in both the **x component** and **y component** edit fields.

Quick Arrow Plots

Click the **Arrow Plot** button on the Plot toolbar to quickly get a plot of that type; this selection disables all other plot types. The plot takes its settings from the **Arrow** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Arrow Plot**.

Setting Arrow Positions

You can plot arrows to visualize a vector field on subdomains or on boundaries. This selection is done in the **Plot arrows on** list. When plotting arrows on boundaries, they are positioned in the midpoint of each boundary element. When plotting arrows on subdomains, the arrows are positioned in a regular grid, selected in the **Arrow positioning** area. Click the button to the left of the **Number of points** edit field to set a spacing using equidistantly distributed arrows. Click the button to the left of the **Vector with coordinates** edit field to explicitly set arrow coordinates using either a space-separated list of coordinates or a vector, for example, 0:10:100. You can specify arrow positioning independently in the x and y directions.

Defining 3D Arrows

To extend an arrow plot to a 3D plot, select the **Height data** check box. Then set the arrow's z -axis base point coordinate using a predefined quantity or an expression.

Defining Arrow Style and Scaling

Use settings in the **Arrow parameters** area to set the arrows' style, scaling, and color:

- Select **Arrow**, **Cone**, or **3D arrow** in the **Arrow type** list to decide which symbol the display should use to draw the vector field.
- Select the **Auto** check box to let COMSOL Multiphysics choose arrow scaling, which ensures that the arrows fit within the grid, or clear it to enter explicit scaling in the edit field.
- Select **Proportional** or **Normalized** in the **Arrow length** list to determine whether the plot gives arrows a length proportional to the magnitude of the vector field or a normalized length (normalized arrows only indicate the field direction).
- Click the **Color** button to set the arrow color; the default is red.

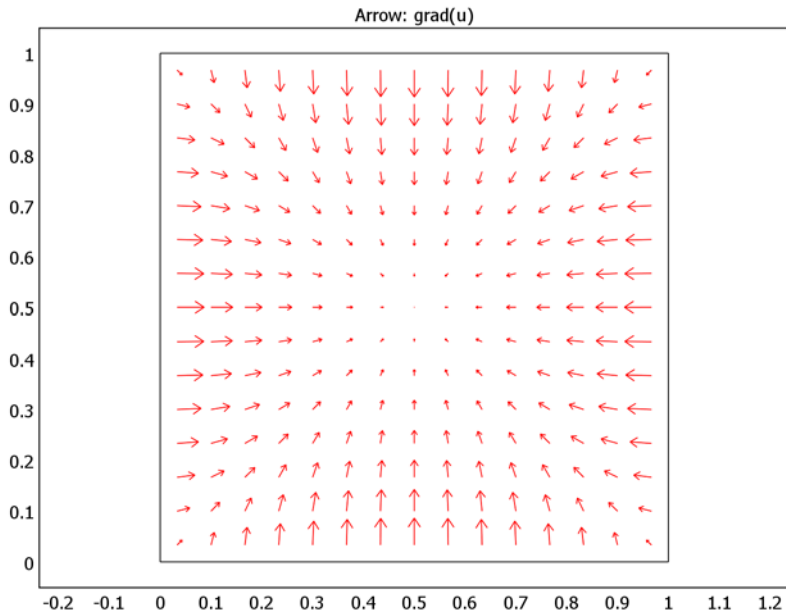


Figure 7-32: The arrows in this plot visualize the gradient of the solution to a 2D Poisson's equations.

2D PRINCIPAL STRESS/STRAIN PLOTS

A 2D principal stress/strain can visualize first, second, and third principal stress or principal strain directions and values. This plot type is available when you use the application modes for structural mechanics in the Structural Mechanics Module, MEMS Module, and Acoustics Module.

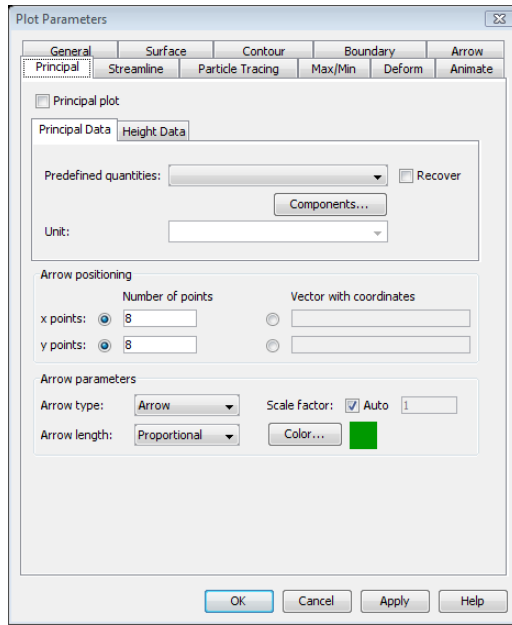


Figure 7-33: The principal stress/strain plot settings in the 2D Plot Parameters dialog box.

The settings are similar to the settings on the **Arrow** page. In particular, the settings in the **Height data** and the **Arrow parameters** areas are analogous to the ones on the **Arrow** page.

To create a principal stress/strain plot:

- 1 Select the **Principal plot** check box to enable this plot type.
- 2 In the **Principal data** area, indicate the quantity that define the lengths and directions of the principal stress/strain symbols. Select from predefined quantities or click **Components** to open the **Principal Components** dialog box in which you can customize the components and values for the principal stress/strain quantity.

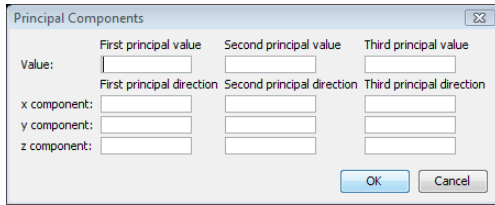


Figure 7-34: The Principal Components dialog box.

Quick Principal Stress/Strain Plots

Click the **Principal Stress/Strain Plot** button on the Plot toolbar to quickly get a plot of that type; this selection disables all other plot types. The plot takes its settings from the **Principal** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Principal Stress/Strain Plot**.

2D STREAMLINE PLOTS

Streamlines visualize a vector quantity on subdomains. A streamline is a curve everywhere tangent to an instantaneous vector field.

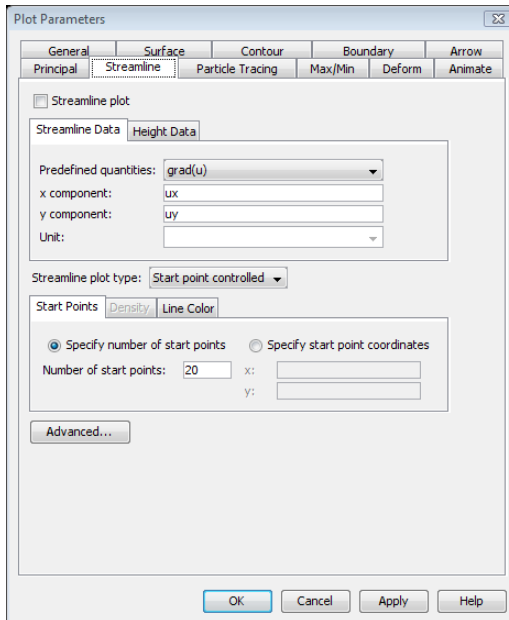


Figure 7-35: The streamline plot settings in the 2D Plot Parameters dialog box.

To create a streamline plot:

- 1 Select the **Streamline plot** check box to enable that plot type.
- 2 Use the settings on the **Streamline Data** page (and, optionally, the **Height Data** page) to specify the vector field. Either select from fields in the **Predefined quantities** list, or enter expressions containing variables in the **x component** and **y component** edit fields.

Quick Streamline Plots

Click the **Streamline Plot** button on the Plot toolbar to quickly get a plot of that type; that selection disables all other plot types. The plot takes its settings from the **Streamline** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Streamline Plot**.

Type of Streamline Plot

The start points for the streamlines can be selected in three different ways:

- They can be specified explicitly by typing in their coordinates. Use the **Start point controlled** setting in the **Streamline plot type** list for this type of visualization.
- They can be randomly selected by the algorithm by specifying the number of start points. Use the **Start point controlled** setting in the **Streamline plot type** list for this type of visualization.
- They can be selected by the algorithm in such a way that the streamlines completely saturates the domain, and are either evenly-spaced or have a density corresponding to the magnitude of the vector field. Use the **Uniform density** or **Magnitude controlled** setting in the **Streamline plot type** list for this type of visualization.

Start Points

Specify the streamline starting points or the number of streamlines using the settings on the **Start Points** page. The option **Specify number of start points** randomly selects the specified number of start points within the geometry. Click the **Specify start point coordinates** button to specify explicit coordinates for the start points in the **x** and **y** edit fields (see “Entering Vector-Valued Expressions” on page 145 for information about valid inputs in each edit field).

Density

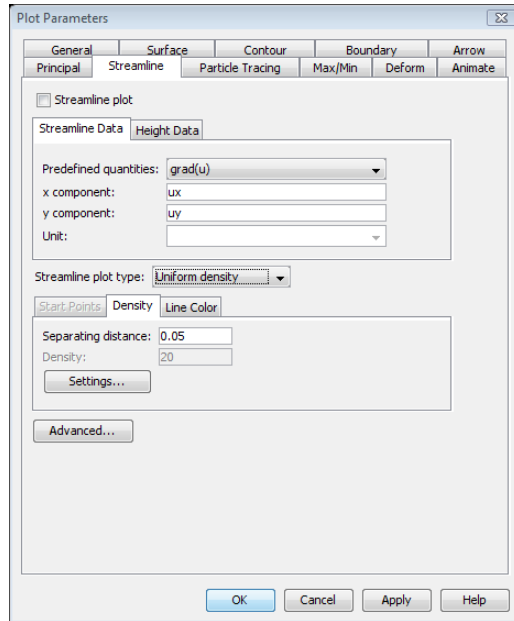


Figure 7-36: The streamline density settings in the 2D Plot Parameters dialog box.

To create streamlines with a uniform density, select **Uniform density** in the **Streamline plot type** list. To create streamlines with a variable density according to the magnitude of the specified vector field, select **Magnitude controlled**.

Uniform density With this setting, the algorithm saturates the entire domain with evenly spaced streamlines. To specify the separating distance between the streamlines, click the **Density** tab, and then enter the distance in the **Separating distance** edit field. The value for the separating distance is a fraction of the mean of the lengths of the bounding box of the geometry. In this case, a streamline stops whenever it gets too close to another streamline or itself (or if any of the general termination criteria specified in the **Advanced Streamline Parameters** dialog box is fulfilled).

Magnitude controlled This setting gives proper streamline plots only for incompressible flow fields. In this case, the algorithm places the streamlines so that the flow between each pair of adjacent streamlines is the same throughout the domain, giving streamlines that are more dense where the magnitude of the field is high. The density is controlled with value in the **Density** edit field. This value is roughly the

number of streamlines. Prior to streamline generation, the software computes a rough estimate of the total flow of the flow field in the model, divides this value with the specified **Density** setting, and uses the resulting value as the flow between each pair of adjacent streamlines. A streamline stops only if it exits the domain or gets too close to its own start point, using the **Loop tolerance** option in the **Advanced Streamline Parameters** dialog box (or if any of the general termination criteria specified in the **Advanced Streamline Parameters** dialog box is fulfilled).

Advanced Density Settings

The **Settings** button on the **Density** page opens the **Advanced Density Settings** dialog box.

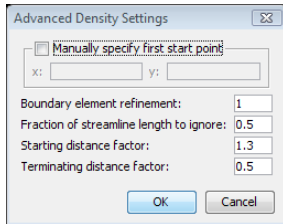


Figure 7-37: The Advanced Density Settings dialog box.

Manually specify first start point By default, the start point for the first streamline is chosen in the element where the highest value of the velocity of the specified vector field occurs. To override this choice, select this option and type in the coordinates.

Boundary element refinement This is a measurement of the density of points on the boundaries used to set up the structure used to measure distances between streamlines. If you experience that streamlines do not behave as expected near boundaries on a coarse mesh, this number might need to be increased. Refining the mesh in the problematic area can also resolve the problem.

Fraction of streamline length to ignore This option is present only when **Uniform density** is selected in the **Streamline plot type** list. When a streamline is close to itself, typically for spiraling streamlines, this number controls how big part of the streamline, starting from its start point, that the streamline itself is allowed to get close to.

Starting distance factor In 2D, this option is present only when **Uniform density** is selected in the **Streamline plot type** list. This is a factor multiplied with the distance specified in **Separating distance** to get the minimum distance between streamlines and the start point for the next streamline. When the domain is close to be saturated with

streamlines, new start points tend to be positioned where the streamline has nowhere to go before it gets too close other streamlines, resulting in short streamlines. A high value of this factor disqualifies the start point and thus reduces the number of short streamlines.

Terminating distance factor This option is present only when **Uniform density** is selected in the **Streamline plot type** list. This is a factor multiplied with the distance specified in **Separating distance** to get the minimum distance between any pair of streamlines. Thus, this distance is the minimal distance under which the integration of a streamline will stop.

Defining 3D Streamlines

Selecting the **Height data** check box on the **Height Data** page extends a streamline plot to a 3D plot by setting the streamline's *z*-coordinate using a predefined quantity or an expression.

Advanced Streamline Settings

Click the **Advanced** button to open the **Advanced Streamline Parameters** dialog box. It contains settings that you typically do not need to change very often. It allows user interaction with the following edit fields and check boxes:

- The **Maximum number of integration steps** edit field; the default is 5000.
- The **Maximum integration time** edit field; the default is infinity (Inf).
- The **Integration tolerance** edit field; the default is 0.01 for 3D and 0.001 for 2D.
- The **Stationary point stop tolerance** edit field; the default is 0.01.
- The **Loop tolerance** edit field; the default is 0.01. This is a fraction of the mean of the lengths of the bounding box of the geometry. If a streamline gets closer to its start point than this distance, the streamline will snap to its start point and thus will be plotted as a connected loop.
- The **Allow backward time integration** check box. This option is present only when you select **Start point controlled** from the **Streamline plot type** list. This option integrates points from the starting points both in the direction of the vector field and in the opposite direction. The default setting allows backward time integration.
- The **Normalize vector field** check box; the default is a vector field for the streamlines that is not normalized.

These settings have the following effects:

- I When calculating streamlines, the software selects a set of starting points (controlled by the streamline start points and the number of start points)

- 2 The algorithm then finds the vectors of the given vector field at these points by interpolation. It normalizes the vector field if you select that option.
- 3 The algorithm integrates the points along the direction of the vector using the integration tolerance using a second-order Runge-Kutta algorithm.
- 4 At the new positions, the algorithm finds vector values by interpolation and performs another integration.

This process stops if:

- It reaches a predetermined number of integration steps (controlled by the maximum number of integration steps entry).
- The points end up outside the geometry.
- The points reach a “stationary point” where the vector field is zero. You control the meaning of “zero” with the stationary point stop tolerance.
- It has used a predetermined amount of “time” for integrating (you control this parameter with the **Maximum integration time** edit field).

Finally, the software connects the calculated points for each streamline consecutively with straight lines.

Note: When integrating, the software uses a pseudotime that has nothing to do with the time in time-dependent problems. Use the massless particle tracing tool to integrate in time-varying fields and to control the real time in stationary fields.

2D PARTICLE TRACING PLOTS

With this plot type, you can visualize *pathlines*, that is, trajectories of particles released in a flow field, which can be time-dependent or static. For time-dependent flows you can 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.

Selecting Particle Type

You can create particle tracing plots for particles with mass or for massless particles. Select **With mass** or **Massless** in the **Particle type** list.

For particles with mass, the software generates the pathlines by solving the fundamental equation of motion

$$m\ddot{x} = F(t, x, \dot{x})$$

for the pathline $x(t)$. Here, m is the particle's mass, F equals the force acting upon the particle, and t is time. This is a system of ODEs for x , which COMSOL Multiphysics solves using a pair of Runge-Kutta methods of orders four and five. The solver advances the algorithm with the solution of order five and uses the difference between the order-five and order-four solutions to obtain the local error estimate.

For massless particles, the equation of motion is

$$\dot{x} = v(t, x)$$

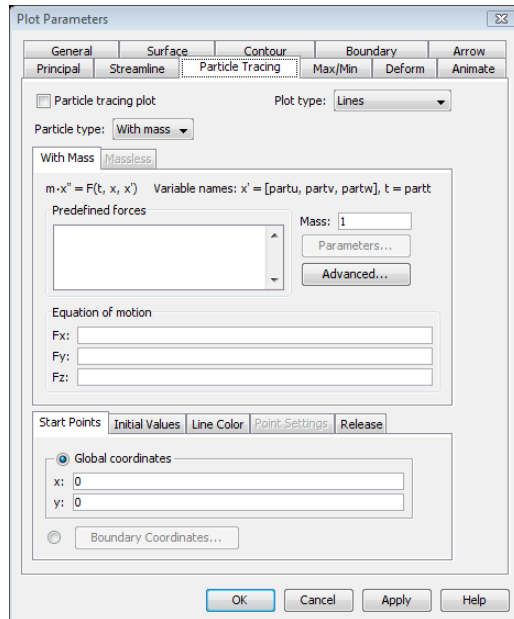


Figure 7-38: The particle tracing plot settings in the 2D Plot Parameters dialog box.

To create a particle tracing plot:

- 1 Select the **Particle tracing plot** check box to enable that plot type.
- 2 Select the plot type. If you select **Lines** in the **Plot type** list, the particle traces appear as lines. Select **Points** to plot points at the endpoints of the pathlines. Select **Points and lines** to plot both lines and points. Finally, selecting **Points along lines** only applies to animations, in which case the final pathlines appear as lines, and the points at the current time appear on top of these lines.

3 Select the particle type in the **Particle type** list.

For particles with mass, use the settings on the **With Mass** page. In the **Predefined forces** area, select from the application mode's predefined particle forces. For examples see “Electromagnetic Force for Particle Tracing” on page 83 or “Khan and Richardson Force for Particle Tracing” on page 152 in the *COMSOL Multiphysics Modeling Guide*. When selecting a predefined force, the corresponding force expression appears in the **Equation of motion** area. Here you can edit the expression or enter a custom force expression. To quickly clear the text fields, use Ctrl-click to deselect all forces in the list of predefined forces. Enter the particle mass in the **Mass** edit field. For some predefined forces, there are additional particle force parameters such as the particle radius. Click the **Parameters** button to define these additional parameters. If the equation of motion depends on the particle's velocity, the top of the **With Mass** page shows the variable names to use for the components of the particle's velocity. Also, for static flows, this area shows the variable name to use for time. You can change these names in the **Advanced Parameters** dialog box; see “Advanced Particle Tracing Settings” on page 522.

For massless particles, use the settings on the **Massless** page. Select a quantity for the velocity field v in the **Predefined quantities** list. The contents of this list depend on the application; typically it is the velocity field or another vector field of interest. You can change the field components by editing the contents in the **x component** and **y component** edit fields. The **Unit** list contains a list of applicable units for the field components.

4 Click the **Start points** tab at the bottom of the dialog box and enter the appropriate coordinates. For information about valid inputs in each edit field, see “Entering Vector-Valued Expressions” on page 145. Click the option button to the left of the **Boundary Coordinates** button; then click that button to open the **Boundary Coordinates** dialog box.

- 5 Select the boundaries where you want your starting points. This is useful, for example, for flow models with one or more inflow boundaries.

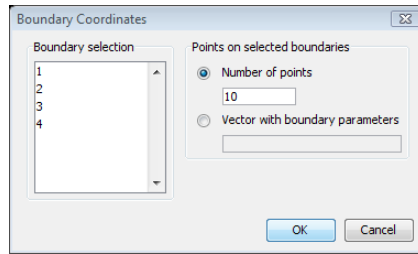


Figure 7-39: The Boundary Coordinates dialog box.

Click the **Number of points** button to use a number of equidistant start points. Click the **Vector with boundary parameters** button to enter a vector of values from 0 to 1 to describe the start points.

Axisymmetric Models

For 2D axisymmetric models, three components for the force are available for particles with mass. When specifying all three, the algorithm solves for a line in 3D in cylindrical coordinates, but the plot only shows the projection on the axisymmetry plane. In this case, the centripetal force is considered; that is, the algorithm solves the equation

$$\ddot{r} = \frac{F_r}{m} + r\dot{\phi}^2 \quad \ddot{\phi} = \frac{F_\phi}{rm} - \frac{2\dot{r}\dot{\phi}}{r} \quad \ddot{z} = \frac{F_z}{m},$$

where m is the particle mass and (r, ϕ, z) are the cylindrical coordinates. Note that the variable corresponding to the velocity component in the ϕ direction (whose default name is `partv`) has the dimension length/time, and equals $r\dot{\phi}$ as $\dot{\phi}$ has the dimension radians/time.

Specifying the Release Time

Click the **Initial Values** tab and enter the time when to release the particles. If you select **Auto**, the first time value that the time-dependent solver returns becomes the start time for time-dependent flows. For stationary flows, COMSOL Multiphysics uses the start time 0. In the **Initial velocity** edit fields type in expressions for the components of the initial velocity. These expressions can include the solution and other postprocessing variables available in the model.

Coloring Settings

On the **Line Color** page you control the coloring of the pathlines. Clicking the **Uniform color** button gives the display a uniform color. To select that color, click **Color** to open the **Pathline Color** dialog box.

It is possible to use an expression to color the pathlines. Click the **Use expression** button, then click the **Color Expression** button. In the **Pathline Color Expression** dialog box you select from predefined quantities or enter an expression. You can also specify a colormap and turn the color scale on or off.

Point Settings

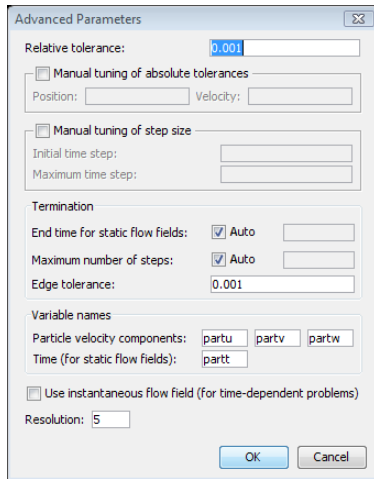
On the **Point Settings** page, set the color and radius scale for the points (used for the **Plot type** settings **Points**, **Points and lines**, and **Points along lines**). The default is to use an automatic scaling of the radius; clear the **Auto** check box to enter another point radius scale factor. From the **When particle leaves domain** list, select the behavior of points that have left the domain. Select **Stick to boundary** to plot the points on the boundary at the exit point, and select **Disappear** to not render these points at all. For static fields, click the **Advanced** button to specify the end time in the **Advanced Parameters** dialog box. 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 you select **Stick to boundary**, and no points appear if you select **Disappear**. To make the points appear, specify an earlier end time.

Starting at Multiple Times

On the **Release** page, select at which times to start the particle tracing algorithm. Click the **Release once** button to start only at the time specified on the **Initial Values** page. Click the **Time between releases** button to specify a time interval between each start time value. Click the **Release at times** button to specify explicit time values at which to start as space-separated values.

Advanced Particle Tracing Settings

Click the **Advanced** button to open the **Advanced Parameters** dialog box.



When solving the 2nd-order ODE $m\ddot{x} = F(t, x, \dot{x})$ for x , the solver first rewrites it as two coupled 1st-order ODEs: one for the position x and one for the velocity \dot{x} , each with two components in 2D and three components in 3D.

The value in the **Relative tolerance** edit field is the relative error tolerance that the ODE solver uses. It applies to all components of the particle's position and velocity. The solver controls the step size so that the estimated error e in each integration step satisfies

$$\begin{cases} e < \max(\text{atolpos}, \text{rtol} \cdot |x_i|) & \text{(for all components } x_i \text{ of } x) \\ e < \max(\text{atolvel}, \text{rtol} \cdot |\dot{x}_i|) & \text{(for all components } \dot{x}_i \text{ of } \dot{x}) \end{cases}$$

where **rtol** is the relative tolerance that you specify, **atolpos** is the absolute tolerance for the particle's position components, and **atolvel** equals the absolute tolerance for the particle's velocity components. You can specify these absolute tolerances in the **Manual tuning of absolute tolerances** area. If you clear the check box at the top of this area, the software automatically computes absolute tolerances for the position and does not use absolute tolerances for the velocity. The automatic absolute tolerance for the position is the mean of the lengths of the bounding box of the geometry multiplied by the relative tolerance. If, on the other hand, you select this check box, you can type in your own values for the absolute tolerances in the **Position** and **Velocity** edit fields. Each

edit field can contain a single value, in which case this absolute tolerance applies to all components of the position or velocity. Alternatively, it can contain a space- or comma-separated list of values specifying individual absolute tolerances for the components of the position or velocity. Thus, in 2D, you can specify two values, and in 3D and 2D axisymmetry, you can specify three values in each of the edit fields.

In the **Manual tuning of step size** area you specify an initial time step and an upper limit of the step size. If you clear the check box at the top of this area, the software automatically computes an initial time step and a maximum time step. To compute the automatic initial time step, the solver uses the initial value of the acceleration (force divided by mass) and the relative and absolute tolerances. The automatic maximum step size is 10% of the total simulation time for time-dependent flows as well as for static flow fields where you specify the end time manually in the **End time for static flow fields** edit field. For static flow fields where you do not set the end time manually, there is no upper limit of the step size. However, in this case, the initial time step is less than or equal to 0.1.

The maximum step size has higher priority than the initial step size; that is, if you specify an initial step size larger than the maximum step size, the solver lowers the initial step size to the maximum step size.

Note that the initial step size, whether you enter it manually or the solver computes it automatically, is not necessarily the first step the solver takes but is rather a first try. If this step leads to an error such that the tolerances are not met, COMSOL Multiphysics lowers it.

If you clear the **Manual tuning of step size** check box, the software shows the automatically computed values for the previous plot in the corresponding edit fields. The value in the **Maximum time step** edit field is the longest time step the solver takes.

The **Termination** area contains settings that determine when to end the particle tracing simulation:

- In the **End time for static flow fields** edit field, specify the maximum time at which to end the particle-tracing simulation for static flow fields. (For time-dependent flows, the automatic end time is the last time that the time-dependent solver returns.) If you select the **Auto** check box, there is no upper limit of the time. In this case, the particle simulation goes on until all particles exit the geometry or the simulation reaches the maximum number of steps. If you activate the **Auto** setting, the edit field contains the time value at which the previous particle tracing plot ended.

- If you specify a limit in the **Maximum number of steps** edit field, the particle simulation ends after this number of steps. If you use the **Auto** setting the limit varies in this way: for static flow fields, the algorithm uses the value 1000; for time-dependent flows, there is no upper limit of the number of steps, and the particle simulation goes on until it reaches the end time. If you activate the **Auto** setting, the edit field displays the number of steps that the previous particle tracing plot took.
- The value in the **Edge tolerance** edit field is an absolute tolerance controlling how close to the geometry boundary the pathlines are cut when they exit the geometry. A lower value cuts the line closer to the geometry boundary.

In the **Variable names** area, specify the names of the particles' velocity components and the time for static flow fields and particles with mass. For static flow fields and massless particles, you can specify the name of the variable for time. Normally there is no need to change the default names. You can use these names in expressions for the force as well as for the color when coloring the pathlines according to an expression.

For time-dependent problems, selecting the **Use instantaneous flow field (for time-dependent problems)** check box freezes the time to the value specified using the **Solution at time** list on the **General** page and considers this a static flow field.

The **Resolution** setting increases the number of output points by the specified factor by adding points between each time step taken in the ODE solver, using a 4th-order interpolation, to produce a smoother output.

Note: The true formulation of Newton's second law of motion is $\frac{d}{dt}(m\dot{x}) = F(t, x, \dot{x})$, that is, you must consider the time derivative of the mass. The particle-tracing algorithm does not solve this equation. Thus, if you specify an expression for the particle mass that depends on time, you will not get correct results.

2D DEFORMED SHAPE PLOTS

It is also possible to visualize a vector field by deforming the geometrical shape. This plot type is primarily intended for visualizing x - and y -displacements (u and v) for problems in structural mechanics. By default, this method automatically scales the deformation to 10% of the geometric domain.

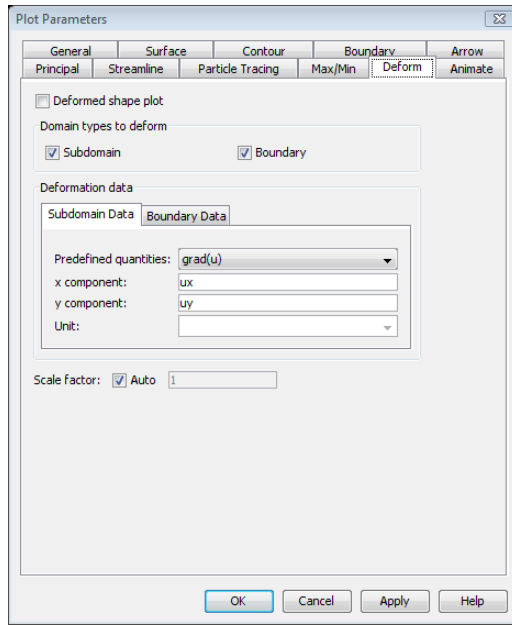


Figure 7-40: The deformed shape plot settings in the 2D Plot Parameters dialog box.

To create a deformed shape plot:

- 1 Select the **Deformed shape plot** check box to enable this plot type.
- 2 Make selection in the **Domain types to deform** area. Select the **Subdomains** check box to deform all subdomain-based plots, and select the **Boundaries** check box to deform the boundary plots.
- 3 Enter the field to deform in the **Deformation data** area. Either select from predefined vector fields, or enter a variable or an expression to define a field's x - and y -components. Click the **Subdomain Data** tab to define deformation data on subdomains; click the **Boundary Data** tab to define deformation data on boundaries.

Scaling the Deformation

You can manually scale the deformation relative to the true deformation by clearing the **Auto** check box and making an entry in the **Scale factor** edit field. If you chose automatic scaling for the previous plot, the field contains the previous calculated scale factor.

2D CROSS-SECTION LINE/EXTRUSION PLOTS

The 2D line/extrusion cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues. The **Line/Extrusion plot** button gets selected when you click the **Line/Extrusion** tab.

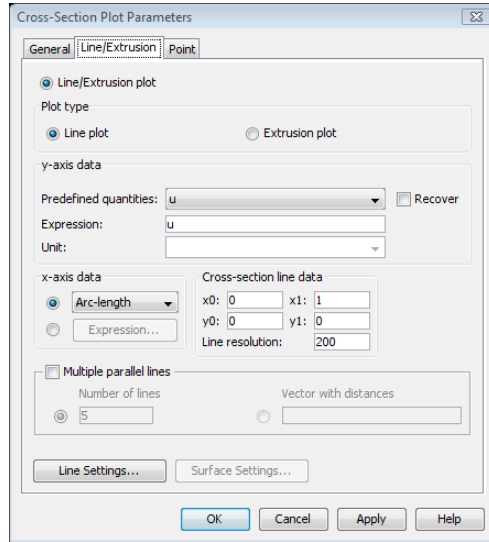


Figure 7-41: The settings for 2D cross-section line and extrusion plots.

Specifying the Cross-Section Line or Extrusion Plot Properties

First specify what the plot should look like. In the **Plot type** area click **Line plot** if you want superimposed line plots, or click **Extrusion plot** to create a continuous surface plot.

Settings in the **y-axis data** area determine the quantity to display on that axis. Either select from predefined quantities or enter an expression containing variables.

Settings in the **x-axis data** area determine the quantity on that axis. To specify the *x*-axis data, click the top button and select the *x*-coordinate, *y*-coordinate, or arc length in the list. Click the bottom button and then click the **Expression** button to enter an arbitrary expression.

In the **Cross-section line data** area enter the 2D coordinates for the plot. Set the start and end point of the line in the **x0**, **y0**, **x1**, and **y1** edit fields. In the **Line resolution** edit field enter the desired number of points (the default is 500).

Select the **Multiple parallel lines** check box to allow visualization on a set of parallel lines. Choose between making an entry in the **Number of lines** edit field or explicitly providing distances from the line specified in **Cross-section line data**. Enter the distances as space-separated values in the **Vector with distances** edit field.

Use the **Line Settings** area to adjust a line plot's color, style, and markers and use the **Surface Settings** dialog box to adjust an extrusion plot's surface.

2D CROSS-SECTION POINT PLOTS

The 2D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. The **Point plot** button gets selected when you click the **Point** tab.

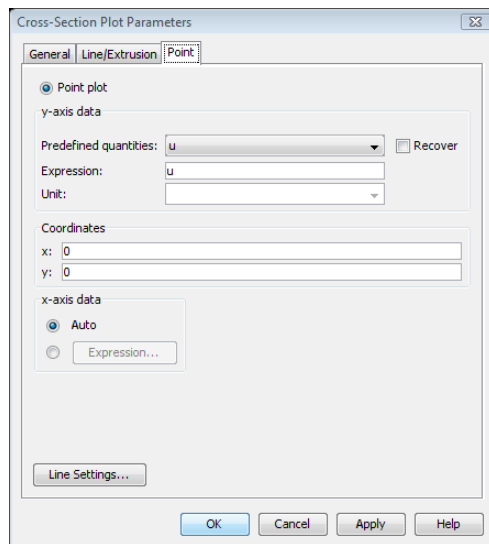


Figure 7-42: The settings for 2D cross-section point plots.

Specifying the Cross-Section Point Plot Properties

The *x*-axis corresponds to time, parameter values, or the eigenvalue number.

The settings in the **y-axis data** area determine the quantity on the *y*-axis. Select from predefined quantities or enter an expression that contains variables.

In the **Coordinates** area enter the x - and y -coordinate values for the plot. Enter the same number of space-separated values in the **x** and **y** edit fields. Alternatively, enter a vector of coordinates, for example, `0:10:100`. For information about valid inputs in each edit field, see “Entering Vector-Valued Expressions” on page 145

Use the **Line Settings** dialog box to adjust the plot’s color, style, and markers.

QUICK 2D CROSS-SECTION LINE AND POINT PLOTS

To get a quick cross-section line plot, click the **Draw Line for Cross-Section Line Plot** button on the Plot toolbar and draw a line in the x - y plane using the mouse.

To get a quick cross-section point plot, click the **Draw Point for Cross-Section Point Plot** button on the Plot toolbar and then click to draw one or more points in the x - y plane. When you have finished, click the button again to create the plot.

The plots appear in a separate figure window and show the currently plotted scalar quantity using surface data, or if it is not available, contour data. If neither is available, the message **No scalar solution data on screen for cross-section quick plot** appears in the message log.

2D Domain Plots

2D DOMAIN SURFACE PLOTS

This plot type visualizes a quantity on one or several subdomains in time, along a parameter range, or for several eigenvalues. The **Surface plot** button gets selected when you click the **Surface** tab.

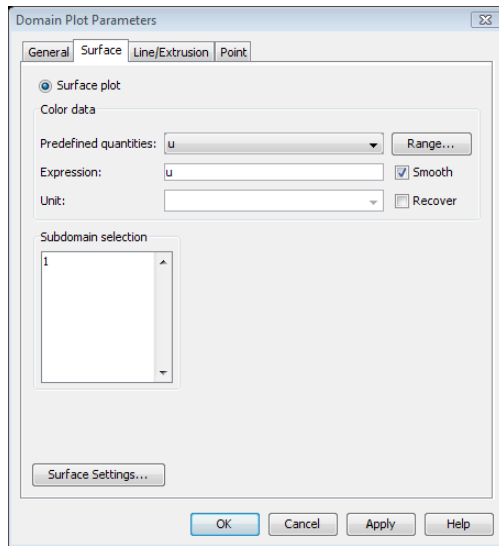


Figure 7-43: The settings for surface plots on 2D domains.

Specifying Domain Surface Plot Properties

Define the surface color in the **Color data** area. Select from predefined quantities or enter an expression containing variables.

Make an appropriate selection for the plot in the **Subdomain selection** list.

Use the **Surface Settings** dialog box to adjust the surfaces' color style.

2D DOMAIN LINE/EXTRUSION PLOTS

This plot type visualizes a solution on one or several boundaries or geometric edges. The **Line/Extrusion plot** button gets selected when you click the **Line/Extrusion** tab.

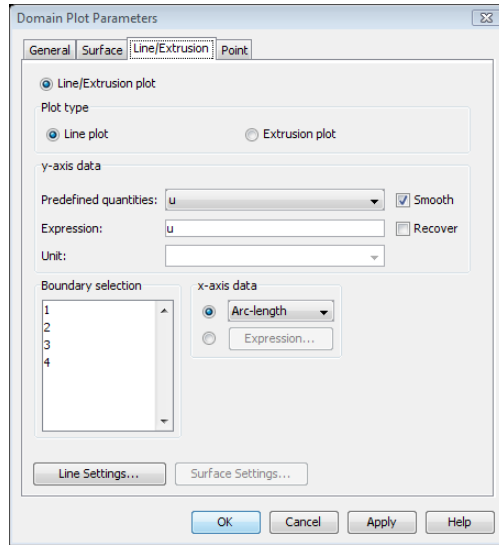


Figure 7-44: The settings for 2D line and extrusion domain plots.

Specifying Line/Extrusion Domain Plot Properties

To show superimposed line plots, in the **Plot type** area, click the **Line plot** button, or click **Extrusion plot** to extrude the data into a continuous surface plot.

Settings in the **y-axis data** area determine the quantity on that axis. Either select from predefined quantities or enter an expression containing variables.

Make an appropriate selection for the plot in the **Boundary selection** list.

In the **x-axis data** area, set the quantity on the x -axis of the plot. Either click the top button and then select to use the x -coordinate, y -coordinate, or arc length; or click the bottom button and then the **Expression** button to specify an arbitrary expression.

Settings in the **Line Settings** dialog box adjust a line plot's color, style, and markers, while the **Surface Settings** dialog box adjusts the surface or the extrusion plot.

2D DOMAIN POINT PLOTS

This plot type visualizes the solution at one or several geometric points or vertices. The **Point plot** button gets selected when you click the **Point** tab.

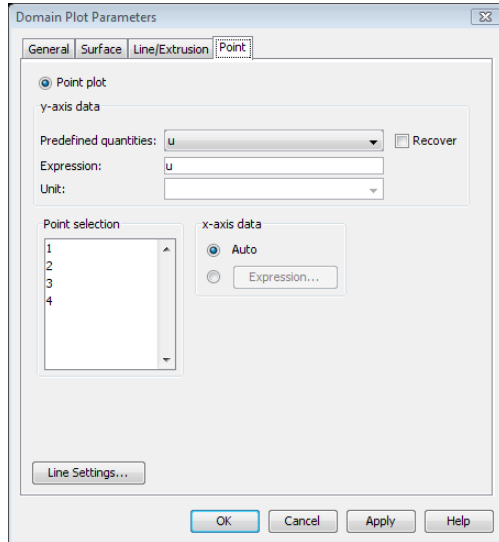


Figure 7-45: The settings for 2D domain plots at points in the geometry.

Settings in the **y-axis data** area determine the quantity on that axis. Either select from predefined quantities or enter an expression containing variables.

Make an appropriate choice for the plot in the **Point selection** list.

Use the **Line Settings** dialog box to adjust the plot's color, style, and markers; use the **Surface Settings** dialog box to adjust the surface or the extrusion plot.

Postprocessing Results in 3D

3D Postprocessing Overview

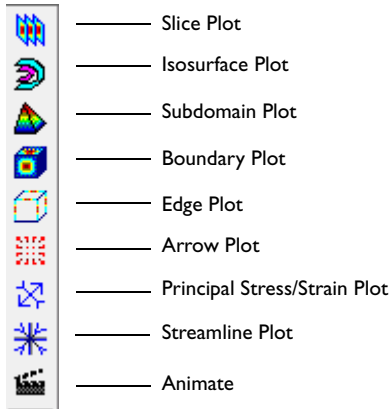
Standard plots in 3D visualize the solution as part of the geometry. It can be a combination of a

- Slice plot
- Isosurface plot
- Subdomain plot
- Boundary plot
- Edge plot
- Arrow plot
- Principal stress/strain plot
- Deformed shape plot
- Streamline plot
- Pathline plots (particle traces)

The software can animate standard plots in time or a along a parametric solution. Cross-section plots can visualize a quantity as a family of plots in an arbitrary plane, line or in an arbitrary set of points in time or along a parametric solution. Domain plots visualize quantities on one or several geometric domains in time or along a parametric solution as a family of plots.

Using the 3D Plot Toolbar

The following plot shows the Plot toolbar for 3D models:



Clicking any of the top eight buttons generates a corresponding plot using the current settings for that plot type in the **Plot Parameters** dialog box. Click the **Animate** button to record and play a movie.

3D SLICE PLOTS

The slice plot is the default type in 3D. It displays a quantity as a set of colored slices through a geometry.

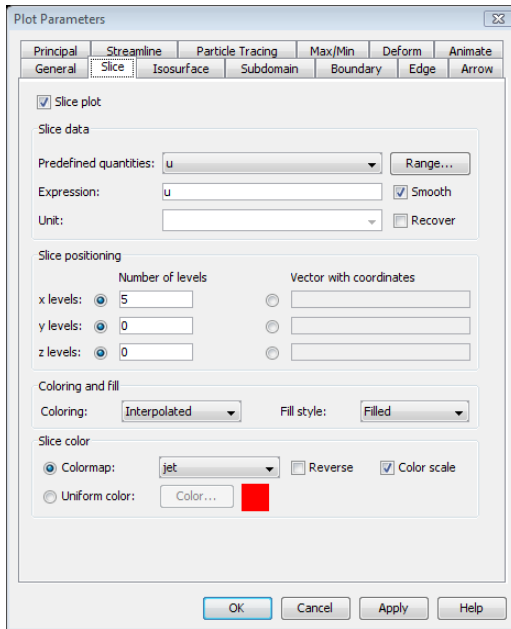


Figure 7-46: The slice plot settings in the 3D Plot Parameters dialog box.

To create a slice plot:

- 1 Select the **Slice plot** check box to enable this plot type.
- 2 Define data for the slice color in the **Slice data** area. Either select from predefined quantities or enter an expression containing variables.

Quick Slice Plots

To quickly get a slice plot, click the **Slice Plot** toolbar button; this action disables all other plot types. The plot takes its settings from the **Slice** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Slice Plot**.

Setting Slice Positions

In the **Slice positioning** area you specify the spacing or positions for the slices. Set slice spacing using edit fields in the **Number of levels** section. Set slice coordinates using the

edit fields in the **Vector with coordinates** section; provide either a space-separated list of coordinates or a vector, for example, 0:10:100. The option buttons to the right of **x levels**, **y levels**, and **z levels** control the desired type of position input.

Choosing Rendering and Fill Style

In the **Coloring and fill** area, choose between interpolated or flat rendering of the surface in the **Coloring** list.

Choose between filled or wireframe rendering with the **Fill style** list. Note that this setting applies to the refined mesh elements for visualization, which can be much denser than the mesh used for computation.

Colormaps and Color Scale Settings

Make an appropriate selection in the **Colormap** list; the default is the **jet** colormap.

Clearing the **Color scale** check box removes the color scale from the display.

Clicking the **Uniform color** button switches the surface to a uniform color, which you select by clicking the **Color** button to open the **Slice Color** dialog box.

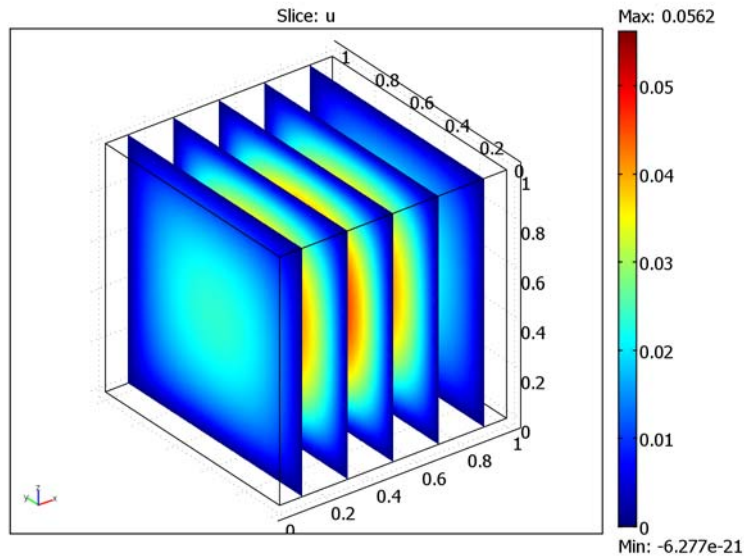


Figure 7-47: A slice plot of the solution to a 3D Poisson's equation. Using transparency would allow you to see through the slices.

3D ISOSURFACE PLOTS

An isosurface plot displays a quantity as a colored set of isosurfaces on which the result has constant value. The plot can also color isosurfaces based on an independent quantity.

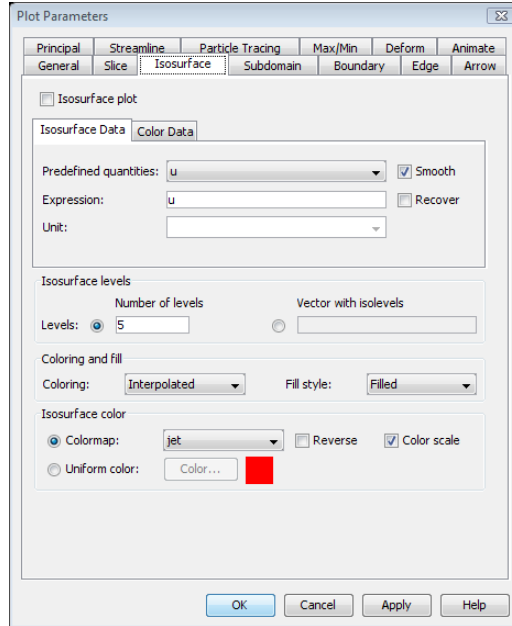


Figure 7-48: The isosurface plot settings in the 3D Plot Parameters dialog box.

To create an isosurface plot:

- 1 Select the **Isosurface plot** check box to enable this plot type.
- 2 In the **Isosurface data** area either select from predefined quantities or enter an expression containing variables. The quantity has a constant value on the resulting isosurfaces.

Quick Isosurface Plots

To get a quick plot, click the **Isosurface Plot** toolbar button; this action disables all other plot types. The plot takes its settings from the **Isosurface** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Isosurface Plot**.

Specifying Isosurface Levels

Make an appropriate entry in the edit fields in the **Isosurface levels** area. You can specify isosurface levels with an integer that indicates the number of isosurface levels (the

default is 5 isosurfaces). Alternatively, enter a vector of levels, for example, 0:10:100. Control which input to use with the **Levels** buttons.

Using Color Data

By default, the isosurfaces take on colors defined by settings in the **Isosurface data** area, and each isosurface has a color constant. Setting the **Color data** check box allows you to color isosurfaces based on an independent value you either select from quantities or specify using an expression.

Colormaps and Color Scale Settings

Make an appropriate selection from the **Colormap** list; the default is **jet**.

Clearing the **Color scale** check box removes the color scale from the display.

Clicking the **Uniform color** button switches the surface to a uniform color, which you select by clicking on the **Color** button to open the **Isosurface Color** dialog box.

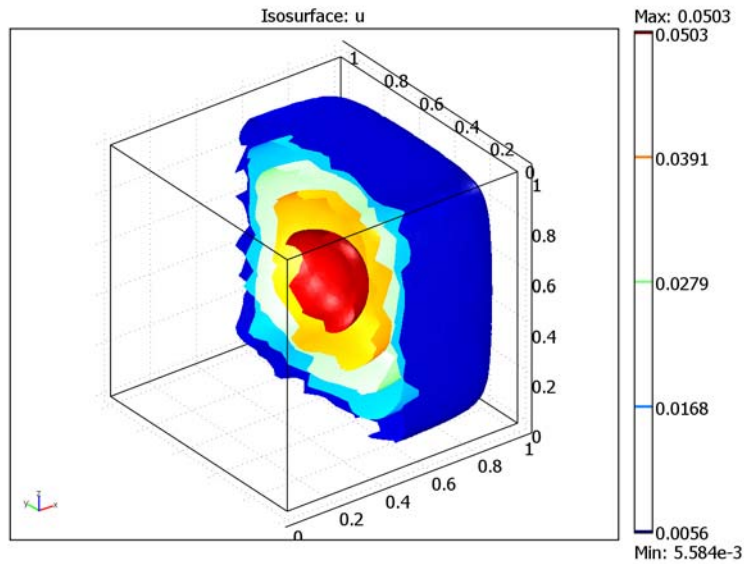


Figure 7-49: An isosurface plot of the solution to a 3D Poisson's equation. The plot uses elements at $x > 0.5$ to get a view of the interior isosurfaces.

3D SUBDOMAIN PLOTS

A subdomain plot displays a quantity defined on subdomains using the subdomain elements. The algorithm renders only their visible surfaces.

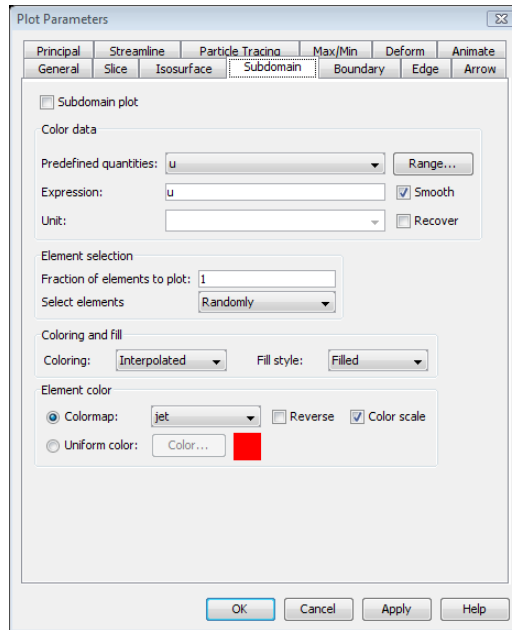


Figure 7-50: The subdomain plot settings in the 3D Plot Parameters dialog box.

To create a subdomain plot:

- 1 Select the **Subdomain plot** check box to enable this plot type.
- 2 In the **Color data** area, either select from predefined quantities or enter an expression containing variables.

Quick Subdomain Plots

To get a quick subdomain plot, simply click the **Subdomain Plot** toolbar button; this action disables all other plot types. The plot takes its settings from the **Subdomain** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots**, and then click **Subdomain Plot**.

Selecting Elements to Plot

In the **Element selection** area you enter a number between 0 and 1 in the **Fraction of elements to plot** edit field. In the **Select elements** list you select either to keep random elements or those with minimum or maximum values.

Coloring Settings

The remaining coloring and fill settings for subdomain plots are similar to those for slice plots; see the section on “3D Slice Plots” on page 534 for more information.

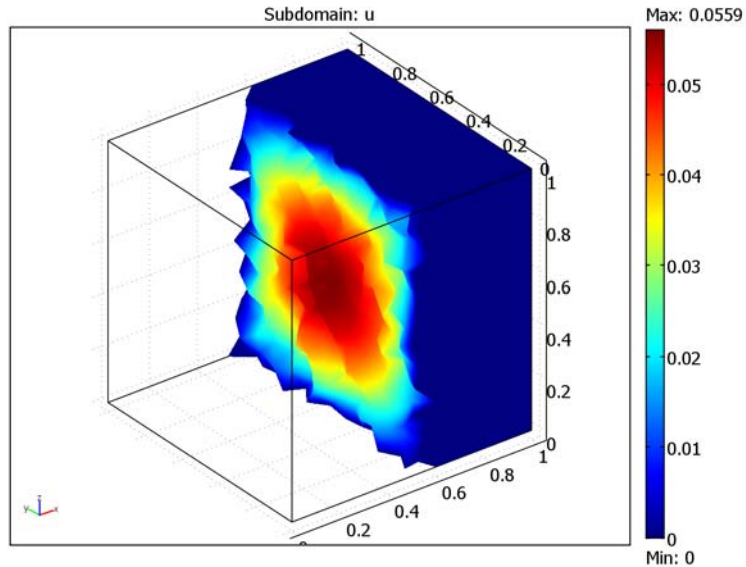


Figure 7-51: A subdomain plot of the solution to a 3D Poisson's equation. The plot only includes elements at $x > 0.5$ to allow a view of the solution at the center of the block.

3D BOUNDARY PLOTS

A boundary plot displays a quantity defined on a boundary.

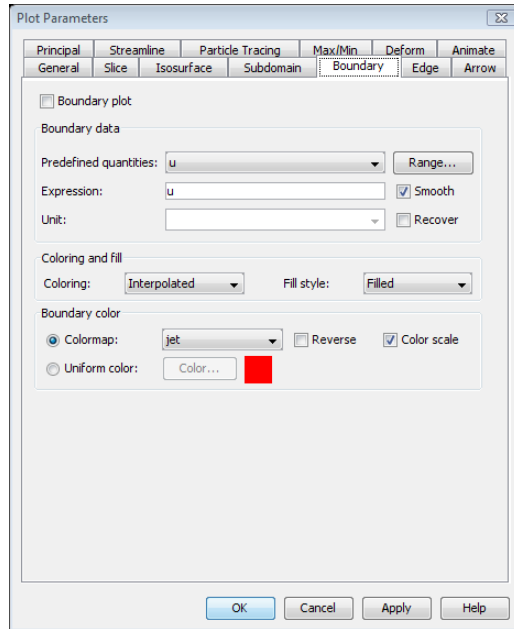


Figure 7-52: The boundary plot settings in the 3D Plot Parameters dialog box.

To create a boundary plot:

- 1 Select the **Boundary plot** check box to enable this plot type.
- 2 In the **Boundary data** area either select from predefined quantities or enter an expression containing variables.

Quick Boundary Plots

To get a quick boundary plot, simply click the **Boundary Plot** toolbar button; this action disables all other plot types. The plot takes its settings from the **Boundary** page. Alternatively, go to the **Postprocessing** menu, point to **Quick Plots** and then click **Boundary Plot**.

Other Boundary Plot Settings

The rest of the settings for boundary plots are similar to those for slice plots; see the section on “3D Slice Plots” on page 534 for more information.

3D EDGE PLOTS

An edge plot displays a quantity defined on edges.

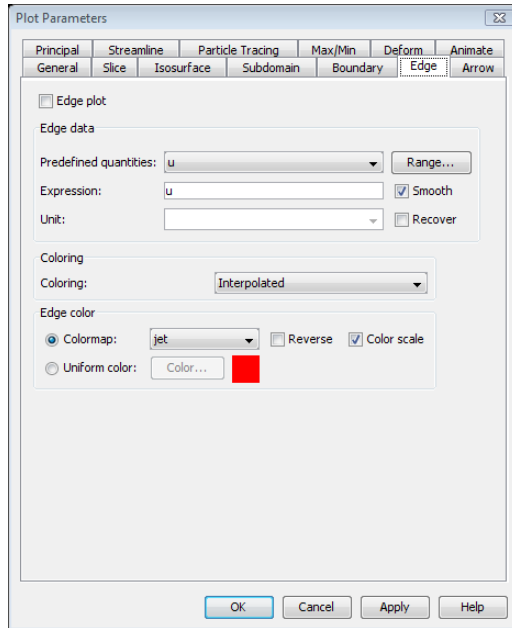


Figure 7-53: The edge plot settings in the 3D Plot Parameters dialog box.

To create an edge plot:

- 1 Select the **Edge plot** check box to enable this plot type.
- 2 In the **Edge data** area either select from predefined quantities or enter an expression containing variables.

Other Edge Plot Settings

The remaining settings on the **Edge** page are identical to those on the **Slice** page.

3D ARROW PLOTS

The arrow plot in 3D is analogous to the 2D case, see “2D Arrow Plots” on page 508, except that it visualizes a 3-component vector quantity and that there is no height data setting. In 3D, you also have the option of plotting arrows to visualize a vector field on the edges of the geometry, in which case the arrows are positioned in the midpoint of each edge element.

3D PRINCIPAL STRESS/STRAIN PLOTS

The principal stress/strain plot in 3D is analogous to the 2D case, see “2D Principal Stress/Strain Plots” on page 510, except that in the 3D case, it is also possible to plot principal stress/strain symbols on boundaries.

3D DEFORMED SHAPE PLOTS

This type of plot visualizes a vector field by deforming the geometrical shape. This type is intended primarily for visualizing x -, y -, and z -displacements (u , v , and w) for problems in structural mechanics. By default, the software scales the deformation to 10% of the problem domain.

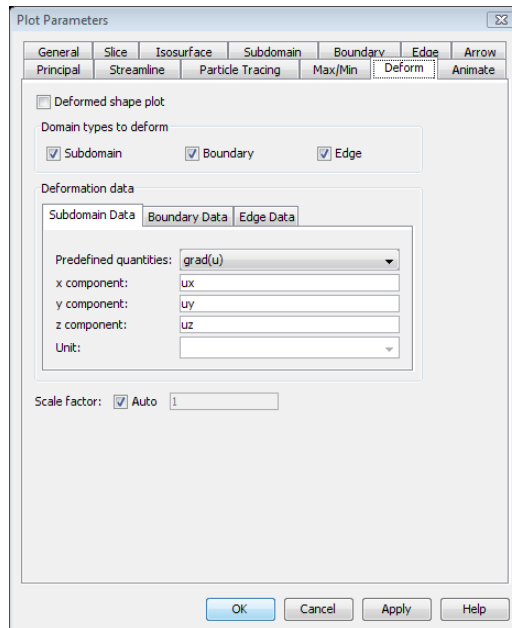


Figure 7-54: The deformed shape plot settings in the 3D Plot Parameters dialog box.

To create a deformed shape plot:

- 1 Select the **Deformed shape plot** check box to enable this plot type.
- 2 In the **Domain types to deform** area select which domain type to deform:
Select the **Subdomains** check box to deform all subdomain-based plots; select the **Boundaries** check box to deform boundary plots; select the **Edge** check box to deform edges.

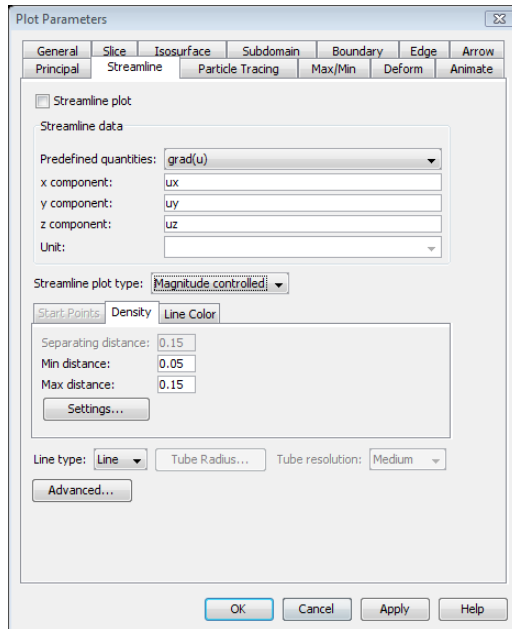
- 3 In the **Deformation data** area choose the field to deform. Either select from predefined vector fields or enter expressions containing a variable or expression defining the field's x -, y -, and z -components.

Scaling a Deformation

You can manually scale the deformation relative to the true deformation by clearing the **Auto** check box and entering a value in the **Scale factor** edit field. If the previous plot used automatic scaling, this edit field contains the last calculated scale factor.

3D STREAMLINE PLOTS

The streamline plot in 3D is analogous to the 2D case (see “2D Streamline Plots” on page 512) except that there is no height data setting. Another difference is the start point selection when **Magnitude controlled** is selected in the **Streamline plot type** list.



Type in a minimum and maximum distance factor. These are specified as fractions of the mean of the lengths of the bounding box of the geometry. The minimum velocity in the model is mapped to the minimum distance and the maximum velocity to the maximum distance. Thus every point on a streamline and on the boundary has a separating distance associated with it. Given a set of streamlines, the start point for the next streamline is selected using these separating distances.

3D PARTICLE TRACING PLOTS

The particle tracing plot in 3D is analogous to the 2D case, see “2D Particle Tracing Plots” on page 517.

3D Cross-Section Plots

3D CROSS-SECTION SLICE PLOTS

A 3D slice cross-section plot visualizes a quantity in one or several planes in time, along a parameter range, or for several eigenvalues.

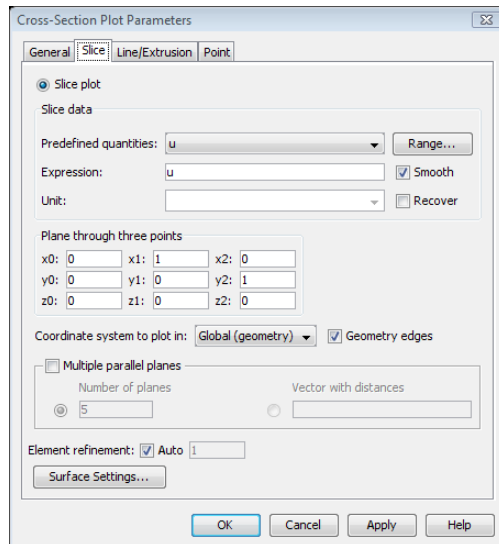


Figure 7-55: The cross-section slice plot settings.

Specifying Cross-Section Slice Plot Properties

In the **Slice data** area either select from predefined quantities or enter an expression containing variables. Select the **Recover** check box to use accurate derivative recovery (see “Accurate Derivative Recovery” on page 469).

In the **Coordinate system** area specify the planes’ orientation and position. Choose between global (in a 3D plot of the entire geometry) and local (in a 2D plot on the cross section) for the embedding of the planes.

Use the edit fields in the **Plane through three points** area to define the location of the plane for the slice plot using the coordinates for three points. If you use the **Draw Plane**

for **Cross-Section Slice Plot** button on the Plot toolbar to draw a plane interactively, COMSOL Multiphysics updates the settings in this area automatically.

Select the **Multiple parallel planes** check box to use more than one plane in the plot. Specify the number of planes either with a numerical entry or by providing the distances to the original plane.

Use the **Surface Settings** dialog box to adjust surface properties.

3D CROSS-SECTION LINE/EXTRUSION PLOTS

A 3D line/extrusion cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues.

The **Line/Extrusion plot** button gets selected when you click the **Line/Extrusion** tab.

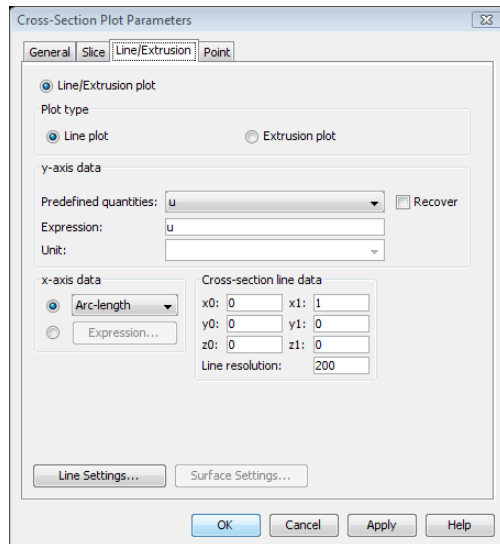


Figure 7-56: The 3D cross-section line and extrusion plot settings.

Specifying Cross-Section Line or Extrusion Plot Properties

In the **Plot type** area, click **Line plot** if you wish the plot to appear as superimposed lines, or click **Extrusion plot** to extrude the line plot into a continuous surface plot.

In the **y-axis data** area you specify the quantity to display on the vertical axis. Either select from **Predefined quantities** or enter an **Expression** containing variables. Select the **Recover** check box to use accurate derivative recovery (see the section “Accurate Derivative Recovery” on page 469).

Settings in the **x-axis data** area determine the quantity on the horizontal axis of the cross-section plot. Click the top button to select the *x*-coordinate, *y*-coordinate, *z*-coordinate, or arc length in the list. Or, alternatively, click the bottom button and then click the **Expression** button to specify an arbitrary expression.

In the **Cross-section line data** area enter the 2D coordinates for the plot. Set the start and end point of the line in the **x0**, **y0**, and **z0** and then the **x1**, **y1**, and **z1** edit fields. Enter a number of points in the **Line resolution** edit field; the default is 500.

Use the **Line Settings** dialog box to adjust a line plot's color, style, and markers; use the **Surface Settings** dialog box to adjust a surface or extrusion plot.

3D CROSS-SECTION POINT PLOTS

A 3D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. The **Point plot** button gets selected when you click the **Point** tab.

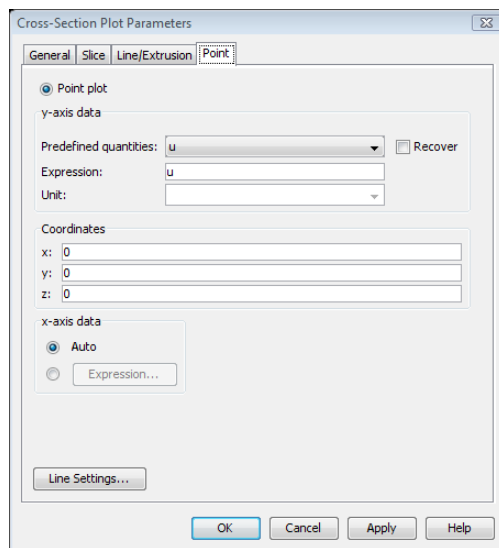


Figure 7-57: The 3D cross-section point plot settings.

Specifying Cross-Section Point Plot Properties

The *x*-axis corresponds to time, parameter values, or an eigenvalue number.

Settings in the **y-axis data** area determine the quantity on that axis. Either select from predefined quantities or enter an expression with variables. Select the **Recover** check

box to use accurate derivative recovery (see “Accurate Derivative Recovery” on page 469).

In the **Coordinates** area provide x -, y -, and z -coordinate values for the plot. Enter the same number of space-separated values in the **x**, **y**, and **z** edit fields. Alternatively, supply a vector of coordinates, for example, 0:10:100.

Use the **Line Settings** dialog box to adjust the line plot’s color, style, and markers.

QUICK 3D CROSS-SECTION FACE AND LINE PLOTS

To get a quick slice cross-section plot, click the **Draw Plane for Cross-Section Slice Plot** button on the Plot toolbar and draw a line with the mouse. This defines a line in the plane of the screen. The software extrudes this line infinitely in the direction of the camera to define a plane that intersects the geometry. This intersection defines the cross-section plane for the plot.

To get a quick line cross-section plot, click the **Draw Line for Cross-Section Line Plot** button on the Plot toolbar and click within the drawing area to draw points in the xy -plane. This action defines points for lines in the plane of the screen. COMSOL Multiphysics extruded these points infinitely to lines in the direction of the camera. The intersection of these lines and the geometry defines the cross-section lines. To remove the points, click the button twice.

The plots show the currently plotted scalar quantity using slice data; or if it is not available, using isosurface data; or if it is not available, subdomain data. If none of these is available, the message **No scalar solution data on screen for cross-section quick plot** appears in the message log.

3D DOMAIN SURFACE PLOTS

This plot type visualizes on one or several surfaces as a quantity in time, along a parameter range, or for several eigenvalues. The **Surface plot** button gets selected when you click the **Surface** tab.

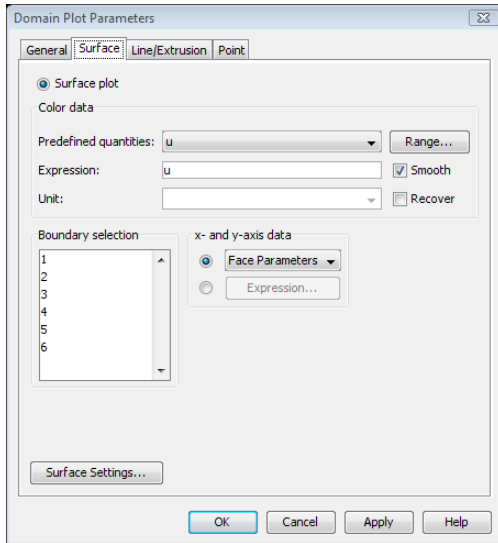


Figure 7-58: The 3D surface domain plot settings.

Specifying Surface Domain Plot Properties

Settings in the **Color data** area determine the quantity for the plot. Either select from predefined quantities or enter an expression containing variables. Select the **Recover** check box to use accurate derivative recovery (see “Accurate Derivative Recovery” on page 469).

Select the boundaries (faces) for the plot in the **Boundary selection** list.

In the **x- and y-axis data** area you define the data on the x - and y -axes. Click the top button to set a predefined quantity. In the list, select face parameters, which uses the local coordinate system for the selected faces, or the xy -, yz -, or zx -plane in the model geometry. Alternatively, click the bottom button and then the **Expression** button to use arbitrary expressions.

The **Surface Settings** dialog box lets you adjust the surfaces’ color style.

3D DOMAIN LINE/EXTRUSION PLOTS

This plot type visualizes the solution on one or more geometric edges.

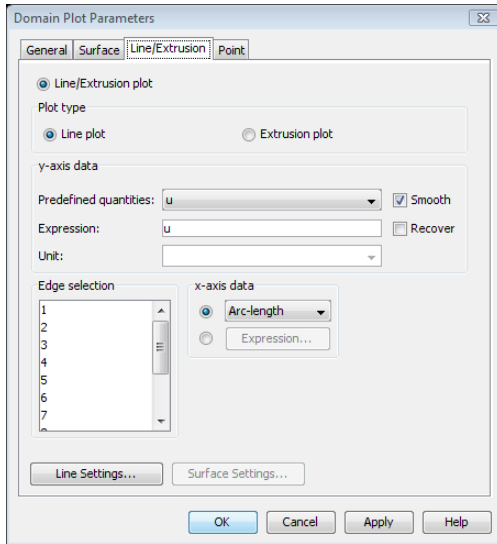


Figure 7-59: The 3D line and extrusion domain plot settings.

Settings on the **Line/Extrusion** page are identical to those in the 2D case; see “2D Cross-Section Line/Extrusion Plots” on page 526.

3D DOMAIN POINT PLOTS

This plot type visualizes the solution on one or more geometric points or vertices.

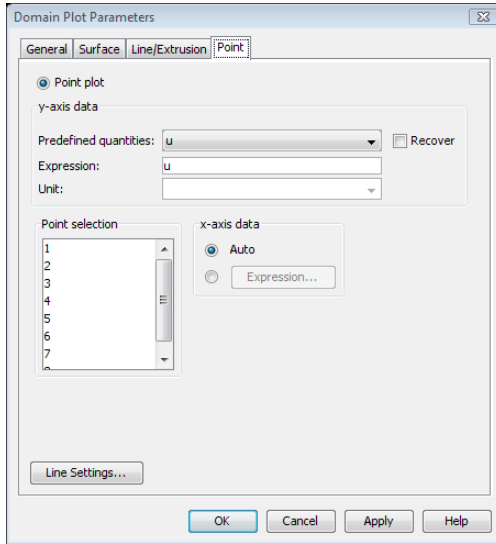


Figure 7-60: The 3D point domain plot settings.

Settings on the **Point** page are identical to those in the 2D case; see “2D Domain Point Plots” on page 531.

The Figure Window

By default, plots you generate using the **Plot Parameters** dialog box appear in the main user interface window, but the **Plot in** list enables you to launch a plot in a separate figure window instead. Plots generated using all other plot dialog boxes always appear in figure windows.

The Figure Window Toolbar

At the top of any figure window is a toolbar with buttons that you can use to process and modify the plot.



The buttons on the toolbar are, from left to right (some are available only for 3D plots):

- **Export Image**—Export the plot in the figure window as an image. This button is the figure-plot equivalent to the menu choice **File>Export>Image** for plots in the main drawing area; see “Exporting Images” on page 493 for details.
- **Print**—Send the plot in the figure window to a printer.
- **Copy**—Copy the plot to the system clipboard.
- **Export Current Plot**—Export data values from the current plot to a text file. This button corresponds to the menu choice **File>Export>Current Plot** for plots in the main drawing area. Therefore, see “Exporting the Current Plot to a File” on page 492 for details. Pure line plots (which can only appear in figure windows) are listed consecutively line after line in the output file.
- **Edit Plot**—Open a dialog box where you can change properties for axes, lines, patches, and so on. See “The Edit Plot Dialog Box” on page 552 for details.
- **Orbit/Pan/Zoom**—Select the general orbit/pan/zoom mode for the mouse in 3D plots: click and move for orbit mode, Ctrl-click and move for pan mode, or Shift-click and move for zoom mode.
- **Pan**—Pan when clicking and moving the mouse.
- **Zoom**—Zoom when clicking and moving the mouse.
- **Dolly In/Out**—Dolly the camera in and out when clicking and moving the mouse.

- **Move as Box**—When this button is selected, a box appears when you click and move the mouse in the drawing area, and no graphics updates are made in real time when changing camera settings. This feature is useful for very large plots or for computers with poor graphics cards.
- **Scene Light**—Turn on all lights added to a plot.
- **Headlight**—Turn on a light mounted on the camera and looking in the direction of the camera.
- **Back in Camera History**—Return to the previous camera position after changing it interactively using the mouse.
- **Forward in Camera History**—Move forward in camera history.
- **Orthographic Projection**—Use orthographic projection.
- **Perspective Projection**—Use perspective projection.
- **Go to XY View**—View the plot in the xy -plane.
- **Go to YZ View**—View the plot in the yz -plane.
- **Go to ZX View**—View the plot in the zx -plane.
- **Go to Default 3D View**—View the plot in the default 3D view.
- **Increase Transparency**—Increase the transparency of patch and surface objects.
- **Decrease Transparency**—Decrease the transparency of patch and surface objects.
- **Zoom In**—Zoom into the plot by a factor of two.
- **Zoom Out**—Zoom out of the plot by a factor of two.
- **Zoom Window**—Zoom in on a rectangular area by clicking and dragging the mouse cursor.
- **Zoom Extents**—Zoom to the extents of the graphics objects currently in the figure window so they appear at the largest possible scale without truncation.

The toolbar for figure windows with 2D line plots additionally contains the following buttons:

- **x-Axis Log Scale**—Use a logarithmic scale for the first axis.
- **y-Axis Log Scale**—Use a logarithmic scale for the second axis.

The Edit Plot Dialog Box

In the **Edit Plot** dialog box you can manually edit many of the properties for the graphic objects in a figure window. To open it, click the **Edit Plot** button on the figure-window toolbar. To the left it shows a tree with the currently defined axes objects in the figure

window along with the graphical objects that lie in each axes object. When you select a node in the tree, a panel appears to the right in which you can change properties for the selected object.

When you select the **Axes** node in the tree, two tabbed pages appear to the right:

- On the **Axis** page you can change axis limits, title, axis labels, and so on. Axis tick marks, the title, and axes labels appear using the font, size, and style you select in the **Font** area.

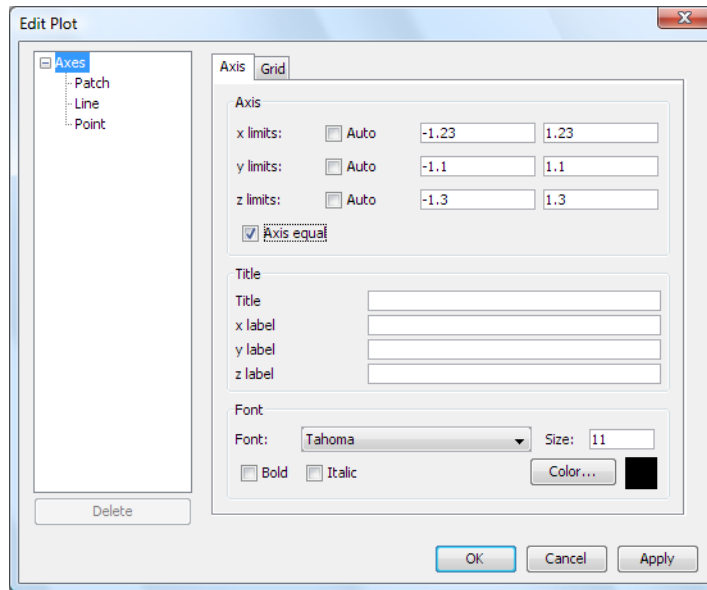


Figure 7-61: The Axis page of the Edit Plot window.

- On the **Grid** page you can choose whether or not to include grid marks separately for each plot axis. To plot without grid lines and tick marks along an axis, clear the **Auto** check box and leave the associated edit field empty. To specify the positions for the grid lines, clear the **Auto** check box and type the desired coordinate values in the edit field.

Figure 7-62 shows the **Edit Plot** dialog box when a **Patch** node is selected in the **Axes** tree. In the **Patch settings** area you can specify the face color, edge color, colormap, and transparency for the selected patch.

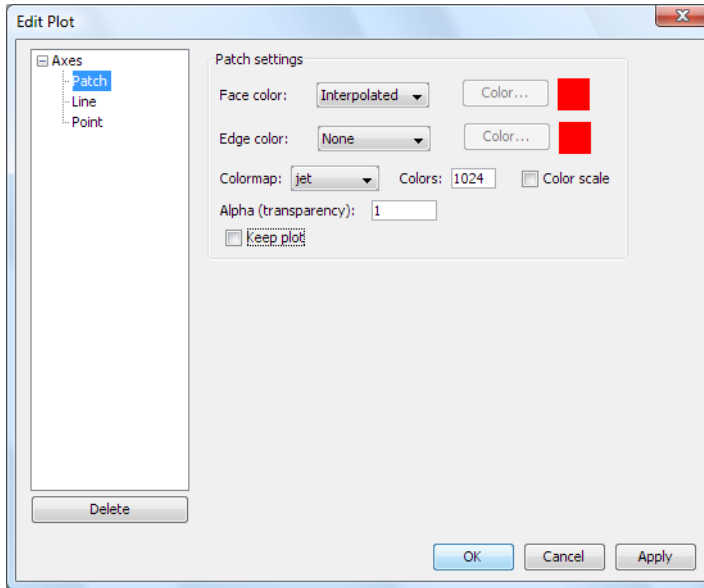


Figure 7-62: The Edit Plot dialog box appearance when a patch object is selected.

Using the Report Generator

Report Generator Overview

The Report Generator is a tool for reporting and documenting your modeling work in COMSOL Multiphysics. It creates a record of the entire model including all the settings you made during the modeling process.

The report this utility generates gives a complete overview of the model, including:

- Model properties
- Geometry
- Physics settings
 - Point settings
 - Edge settings
 - Boundary settings
 - Subdomain settings
 - Other physics settings
- Mesh
- Solver parameters
- Visualization of the solution

During the modeling process, the Report Generator provides an overview of all work completed thus far. The report includes all settings you have made in the various dialog boxes across all application modes and geometries.

These reports are easy to publish either as printed hard copies or as electronic documents suitable for the Internet. The utility generates the HTML report in a file format that makes it possible to customize the report in Microsoft Word or any HTML editor.

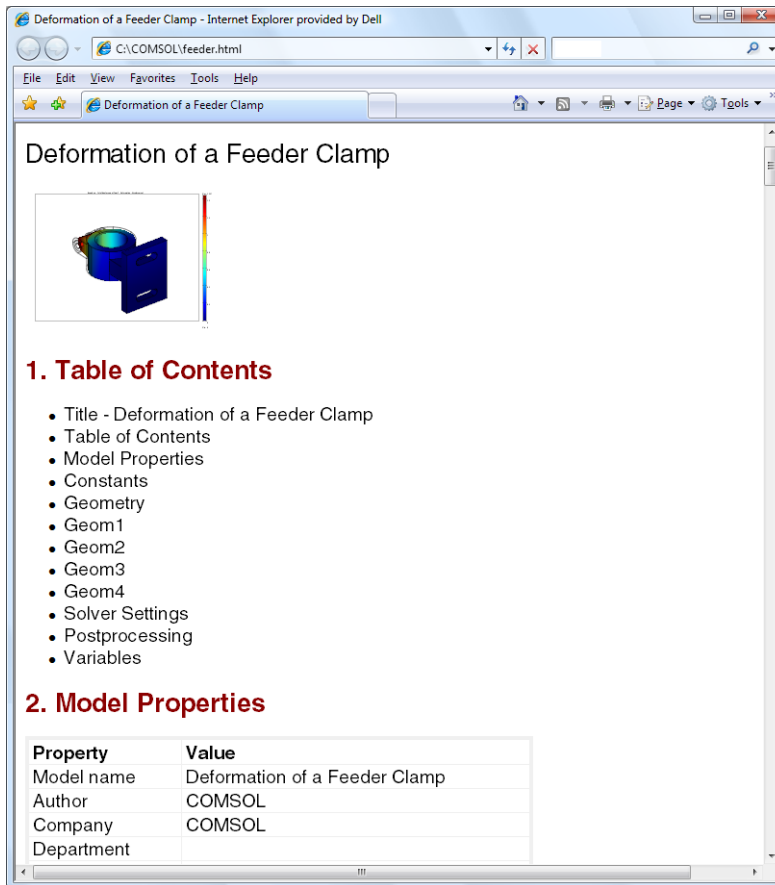


Figure 7-63: An example report. The first page contains model properties and a model description.

Creating a Report

To make a report, go to the **File** menu and choose **Generate Report** (or press Ctrl+G). There are two choices for the output format:

- Direct printing
- HTML

DIRECT PRINTING

To send the report directly to a printer, you must have a printer installed on your computer system. You select the paper type, either letter size (8.5 inches-by-11 inches) or A4 (210 mm-by-297 mm) from the **Output format** list. Click **OK** to start the report printing.

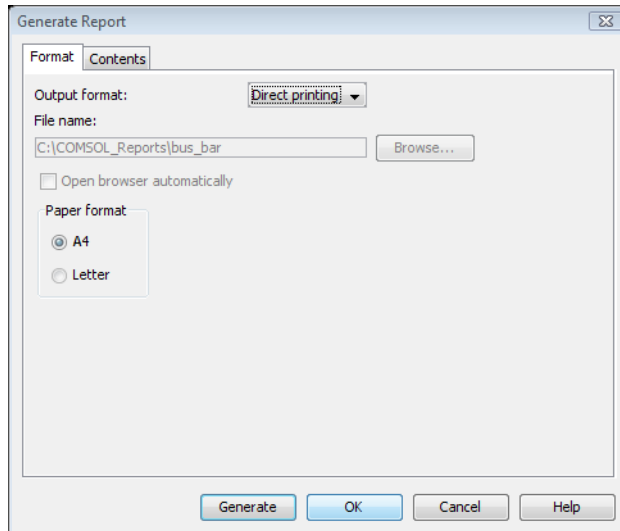


Figure 7-64: The Generate Report dialog box with settings for direct printing.

The Report Generator first scans the model to create plots for the report. During that stage, COMSOL Multiphysics displays a Progress window that shows how far the Report Generator has completed image preparation (see Figure 7-66 on page 559) and then opens the **Print** dialog box, where you control the printer selection and settings.

GENERATING AN HTML REPORT

When generating an HTML report, you specify its name and where to store the file. Do so either by entering the path and file name or by clicking **Browse** in the **Select HTML Output File** dialog box and browsing to the desired location. If you point to an existing file, the Report Generator warns you before overwriting it. If you specify only a name, the software generates no warning and overwrites any existing files.

Note: The Report Generator overwrites the HTML and associated image files if you reuse the same report name and path. To avoid overwriting a previous report, change the name or path of the HTML report.

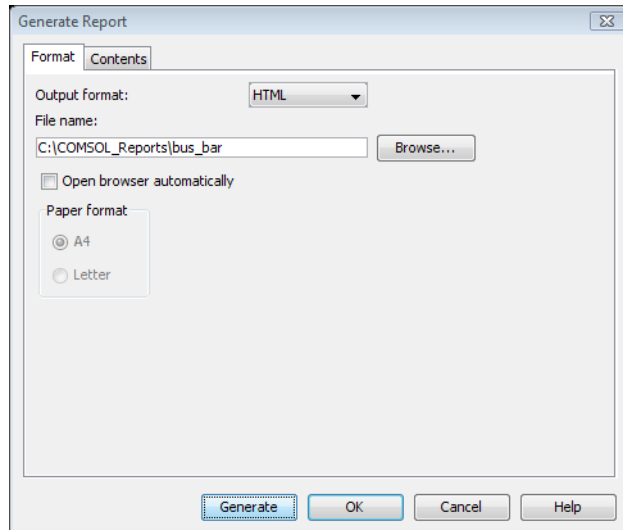


Figure 7-65: The Generate Report dialog box with settings for a report in HTML format.

The Report Generator stores the report in a file with the chosen name and gives it the extension `.html`. It stores images included in the report in a subdirectory with the same name as the report plus the suffix `_rep`.

Select the **Open browser automatically** check box to view the report in a web browser as soon as it is available.

REPORT GENERATOR PROGRESS WINDOW

When you start the Report Generator, COMSOL Multiphysics displays a Progress window where you can monitor the utility's progress as it creates the report. You can stop report generation at any time by clicking **Cancel**. The Progress window normally closes when the report is ready, but you can keep it open by clearing the **Close automatically** check box. After printing the report or generating an HTML file, the

Report Generator prints a confirmation into the message log at the bottom of the user interface.

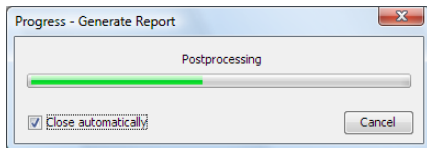


Figure 7-66: The Progress window during report generation.

PUBLISHING AND EDITING OF THE REPORT

Once you have created an HTML report, you can publish or edit it in any HTML editor such as Microsoft Word or a dedicated HTML editor. If you want to publish a report on the Internet or send it to a colleague, be sure to send the HTML file as well as the folder with the images.

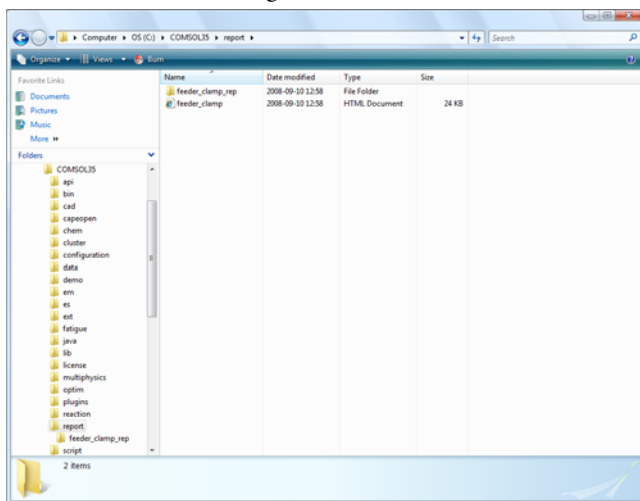


Figure 7-67: To publish an HTML report, include both the HTML file along with the associated image folder.

Report Contents

This section gives a short description of a report's contents and also outlines its structure.

A report summarizes all the actions that you have taken up to the point of generating the report, thus showing a snapshot of the modeling process. You can create a report

at any stage in the modeling process to get an overview of the work thus far and then continue modeling.

Note: If you compute a solution and then change any settings in the model without re-solving it, the report includes the new settings but retains the old solution and postprocessing data. The Report Generator always creates reports based on the models in their present state.

A report has three main sections:

- Introduction and model properties
- Geometry and physics settings
- Solution and postprocessing

Each section contains several subsections including all the model information.

INTRODUCTION AND MODEL PROPERTIES

The first section, the report introduction, includes model properties and other definitions such as constants that are common throughout the different geometries in the model.

GEOMETRY AND PHYSICS SETTINGS

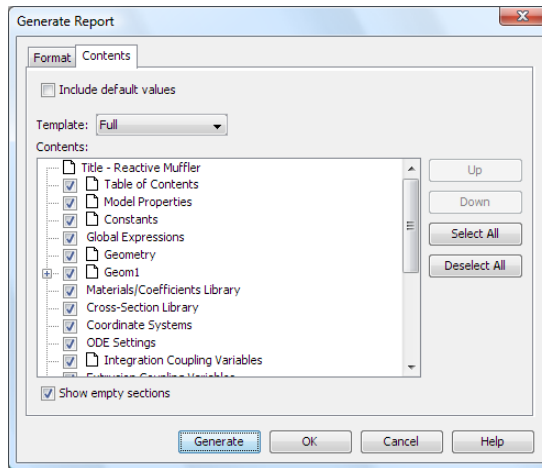
The second section, with physics settings, repeats for each geometry in the model. In each case it lists the physics settings for each application mode that is active in that portion of the geometry. This section also includes a description of each application mode. It prints in bold font all physics settings that you have changed and thus differ from the default values.

SOLUTION AND POSTPROCESSING

The last section contains the materials libraries, functions, and solver settings used to solve the model. The final chapter reviews postprocessing results. The solution and postprocessing sections are common to the entire model.

CHANGING THE CONTENTS OF THE REPORT

To change the contents of the report, click the **Contents** tab in the **Generate Report** dialog box.



The **Contents** area shows which sections the report includes and the order in which they appear.

If you select the **Include default values** check box, the report also includes the default values for all physics settings.

There are four templates for the report format that you can choose in the **Template** list:

- **Full**: the default, which generates a full report
- **Summary**: a summary of the model, including physics settings and postprocessing but without table of contents and index
- **Settings**: similar to the Summary template but includes a table of contents, an index, and all variables
- **None**. This includes all other combinations. You can customize the contents by selecting or clearing the check boxes for each section.

Select the **Show empty sections** check box to display empty sections, that is, sections that have no contents for the current model. If you select the Full report template or the **Show empty sections** check box, you can use the **Up** and **Down** buttons to move the sections in the report.

OVERVIEW OF THE SUBSECTIONS

A complete list of the subsections that the Report Generator can produce follows. A report of a COMSOL Multiphysics model contains any relevant subset of this list.

- Model properties, description and summary
- Constants
- Geometries (the report includes the following sections for each geometry)
 - Draw mode
 - Point mode
 - Edge mode
 - Boundary mode
 - Subdomain mode
 - Physics (for each geometry and each application mode)
 - * Scalar expressions
 - * Point expressions
 - * Edge expressions
 - * Boundary expressions
 - * Subdomain expressions
 - * Mesh and mesh cases
 - * Application mode
 - * Scalar variables
 - * Application mode properties
 - * Variables
 - * Point settings
 - * Edge settings
 - * Boundary settings
 - * Subdomain settings
- Materials libraries
- Functions
- Interpolation functions

- Solver settings
 - Linear system solver
 - Stationary solver
 - Time-dependent solver
 - Eigenfrequency solver
 - Parametric solver
 - Optimization/sensitivity solver
 - Adaptive solver
 - Advanced solver settings
- Postprocessing
 - Eigenvalues or eigenfrequencies
- Equations
- Variables

The report displays default settings in a regular font, and it lists in bold font any physics settings you have changed from the defaults. It leaves blank any settings that are inactive.

Model Properties, Description, and Summary

In COMSOL Multiphysics it is possible to set some general model properties that in turn appear at the beginning of a report. These properties come from the **Model Properties** dialog box, which you open by going to the **File** menu and choosing **Model Properties**. See “Setting Model Properties” on page 571 for more information about the model properties. The Report Generator always includes certain model properties at the beginning of a report:

The report includes information from the **Info** tab in the **Model Properties** dialog box to show the COMSOL Multiphysics version as well as the modules and application modes with which the model was built.

Next it includes a separate section for each tab:

- **General:** Information about the model’s author and possibly a URL to a location where the reader can find more information about the model
- **Description:** An introductory description of the model
- **Notes:** This property can be used, for example, to accommodate comments on the modeling results

Geometries

All geometries that exist in a model appear in the report.

If a model includes multiple geometries, the report includes information about all geometries, including all application modes for each geometry.

Point, Edge, Boundary, and Subdomain Mode

For each mode, the report shows the geometry and labels for the points, edges, boundaries, and subdomains. The report uses these later when describing the corresponding settings for the included application modes.

Point Settings

For each application mode, the report lists the point settings by groups with identical settings. It shows in bold font any settings that differ from their default values.

Edge Settings

For each application mode, the report lists the edge settings by groups with identical settings; note that edge settings exist only in 3D models. It shows in bold font any settings that differ from their default values.

Boundary Settings

For each application mode, the report lists the boundary settings by groups with identical settings, but only those boundaries active in the geometry. It shows in bold font any settings that differ from their default values.

Subdomain Settings

For each application mode, the report lists the subdomain settings by groups with identical settings. It shows in bold font any settings that differ from their default values.

Mesh

To document the mesh, the report includes a plot of the mesh along with mesh statistics. If you solved the model using mesh cases, the report includes a plot and statistics for each mesh case.

Materials Libraries

If the model uses a materials library, the relevant materials appear in the report.

Solver Settings

The report lists all solver settings, and when applicable it also includes a solver script. For the multigrid solver, the report lists all relevant settings for:

- Preconditioner

- Presmoothing and postsmoothing
- Coarse solver

Note: The report includes only the current solver settings. If you solve the model in steps, we recommend you use solver scripting because the report contains the solver script, and that script provides a documentation of your solution steps.

Postprocessing

To document model results, the Report Generator always includes the final postprocessing plot. If you want more postprocessing plots in the final report, you must copy/paste them there using functions on the Edit menu.

In addition, the report includes any eigenvalues or eigenfrequencies depending on the settings in the Application Mode properties.

Note: The report includes the last postprocessing plot the software created—even if you have changed the settings and have not re-solved the model.

Model Maintenance and Model Libraries

This chapter describes how to save and open COMSOL Multiphysics models. You also find a description of the different model formats. In addition, you find a list of models in the COMSOL Multiphysics Model Library and descriptions on how to create your own model library.

Opening and Saving COMSOL Multiphysics Models

Model Files Formats

There are two different file formats for COMSOL Multiphysics models:

- **Model MPH-file**—the default file format
Consists of a binary part containing data such as the mesh and the solution, and a text-file part with other model information. Model MPH-files have extension `.mph`. Earlier versions of COMSOL Multiphysics (FEMLAB) used the file extension `.fl`. See “Model MPH-files” on page 570 for more information.
- **Model M-file**—editable script file for COMSOL Script or MATLAB
Contains a sequence of COMSOL Multiphysics commands. Run the Model M-files in COMSOL Script or in MATLAB or open them in COMSOL Multiphysics. You can edit the file in a text editor to add additional commands. See “Model M-Files” on page 574 for more information.

Opening a COMSOL Multiphysics Model

To open a saved COMSOL Multiphysics model, follow these steps:

- 1 From the **File** menu, choose **Open**.
- 2 From the **Files of type** list in the **Open Model** dialog box, choose the file type that matches your model. Use the **Look in** list to browse and select the file you want to open.
- 3 Click **Open** to open the model.

When the **Preview** button in the **Open Model** dialog box is pressed in, the dialog box shows the model image and model description, if they exist (for more information about saving model images and model descriptions, see “Setting Model Properties” on page 571). Click the **Preview** button to show or hide the preview.

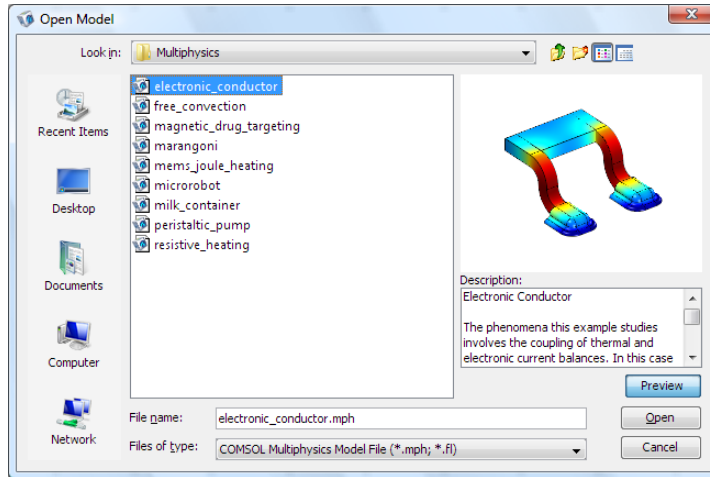


Figure 8-1: The Open Model dialog box with preview.

On Windows you can also double-click the model's icon to open the model in COMSOL Multiphysics. Note that by doing so, the folder where the model file (MPH-file) is located becomes the current working directory in COMSOL Multiphysics.

THE PROGRESS WINDOW

When COMSOL Multiphysics opens a Model MPH-file, you can follow the progress in the Progress window. For some large models, plotting the results can take a long time (see “Opening Models Faster Using Rough Plots” on page 472 on how to speed up the loading of such models). Click **Cancel** in the Progress window if you want to cancel the opening of a model.

The Progress window normally closes automatically. If you want it to stay open, clear the **Close automatically** check box.

Saving a COMSOL Multiphysics Model

To save your COMSOL Multiphysics models, follow these steps:

- 1 From the **File** menu, choose **Save As**.

If the model you are working on has never previously been saved, you can alternatively choose **Save**. If, on the other hand, the model has been saved before,

the latter command updates the file while keeping the current name and format, and the following steps do not apply.

- 2 In the **Files of type** list, select the file type that you want to use.
- 3 Specify the folder using the **Save in** list. Click on a file to save the current model to an existing file, or type in the file name in the **File name** edit field.
- 4 Click **Save**.

Saving on Server Computer

When running a COMSOL Multiphysics server on a remote computer you can save the Model MPH-file both on the client computer and on the server computer. The save on the client use the normal **Save As** dialog box. Do the following steps to save on the server computer instead:

- 1 Click the **Save On Server** button, which appears in the **Save As** dialog box. This button is only available when running a remote server.

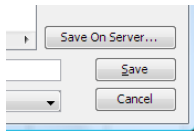


Figure 8-2: The *Save On Server* button (figure shows part of dialog box only).

- 2 A new dialog box now opens. Enter the full path to the file you want to save in the edit field. Provide the path in the server computer's file system. Then click **Save** to save the file.

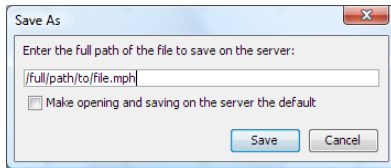


Figure 8-3: The *Save As* dialog box for saving on the server.

Model MPH-files

The Model MPH-files, with extension `.mph`, are the standard files to use when you save your COMSOL Multiphysics models. The file contains both binary and text data. The mesh and solution data are stored as binary data, while all other information is stored as plain text.

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

Setting Model Properties

You can set properties for the model using the **Model Properties** dialog box. Open the dialog box by choosing **Model Properties** from the **File** menu.

ADDING MODEL INFORMATION DATA

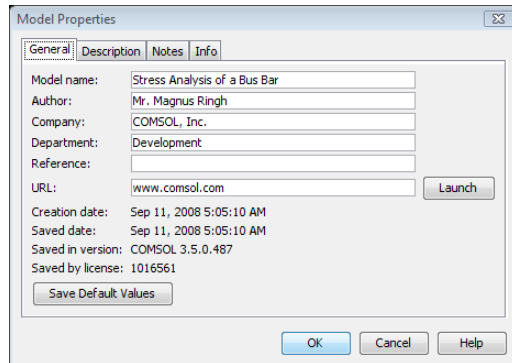


Figure 8-4: The Model Properties dialog box.

The **Model Properties** dialog box provides the ability to save the following information with a model:

- A model name, which need not be the same as the file name
- Author's name
- Company name
- Department
- Reference, which could be a reference to a paper or book from the scientific literature
- A URL (web address) that contains information related to the model. Click the **Launch** button in the dialog box to open a browser and display the web page. COMSOL Multiphysics adds `http://` to the beginning of the URL if it is missing.

You can specify all these properties except the model name as being default values for all models. To do so, fill out the fields and then select the **Set as default values** check box and click **OK**.

In addition, the **Model Properties** dialog box contains information about the creation date and the date when you last saved the model as well as the COMSOL Multiphysics version and license number.

ADDING MODEL DESCRIPTIONS AND NOTES

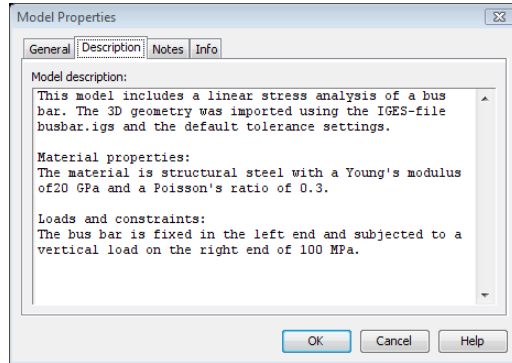


Figure 8-5: The Model description field, where you can enter information about the model.

To enter a short summary of the current model, click the **Description** tab and enter appropriate text in the **Model description** edit field before saving the model. COMSOL Multiphysics saves the text with the model and displays it in the **Model Navigator** when you select that model.

To add other notes such as results data and a discussion, click the **Notes** tab and enter any desired information in the **Results description** edit field.

MODEL INFORMATION

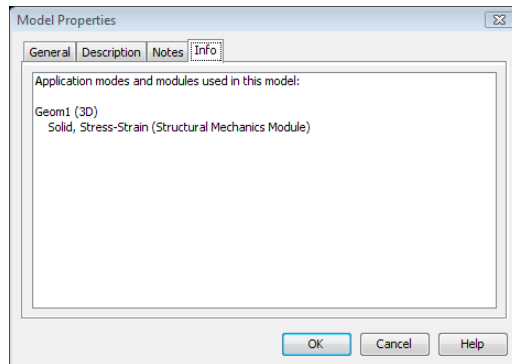


Figure 8-6: The Info Page shows the geometries and application modes in the model.

To get information about which application modes and modules the model uses, click the **Info** tab. COMSOL Multiphysics displays this information for the current model (you cannot edit this field's contents).

SAVING A MODEL IMAGE

To illustrate the model, you can save a model image that will be displayed in the **Model Navigator**. To save the current COMSOL Multiphysics plot as a model image, choose **Save Model Image** from the **File** menu. The model image is a copy of the current plot.

Model M-Files

You can save an entire modeling session as a sequence of commands. This command sequence is called a *Model M-file*. You can:

- Run these commands directly from the COMSOL Script or MATLAB command line or load the file into COMSOL Multiphysics
- Modify the Model M-file using COMSOL Script or MATLAB commands

If you want to open a modified Model M-file in COMSOL Multiphysics, it is important to follow the syntactic rules outlined in the *COMSOL Multiphysics Scripting Guide*.

Note: You can always load a Model M-file into COMSOL Multiphysics. To run the Model M-file from a command line, use COMSOL Script or MATLAB. Check the *COMSOL Installation and Operations Guide* for specific system requirements.

Using Model M-files

The Model M-file is useful for many reasons. For example, it makes it possible to:

- Easily document modeling in a readable format
- Work with COMSOL Multiphysics command-line functions in COMSOL Script or MATLAB
- Save, modify, and load Model M-file back into the COMSOL Multiphysics user interface
- Insert COMSOL Script or MATLAB commands anywhere in the Model M-file
- Start modeling in the COMSOL Multiphysics user interface, then save it as a Model M-file and continue the modeling in the COMSOL Script or in the MATLAB environments with a starting point that is a Model M-file with the correct syntax for reproducing the model that you created in the COMSOL Multiphysics user interface.

The Model M-file records your modeling session to a text file. This script contains all changes to the model. Using a text editor, you can manually edit, add, and remove commands from the Model M-files.

MODIFYING MODEL M-FILE CONTENTS

You can modify some of the contents in the Model M-file to get a more or less compact representation of the model.

Including Default Values

By default, the Model M-file does not include fields with their default values; that is, values that you have not changed when creating the model. To include also the default values in the generated file, do as follows:

- 1 From the **Options** menu, open the **Preferences** dialog box.
- 2 Click the **Modeling** tab.
- 3 Select the **Include default values** check box in the **Model M-file/FEM export** area.
- 4 Click **OK**.

Note: This setting only affects the commands for subsequent modeling steps. It does not include default values for modeling steps that COMSOL Multiphysics has already recorded.

Shrinking Coefficients

By default, COMSOL Multiphysics “shrinks” the coefficients in the physics settings to a more compact notation than the full syntax, which makes the Model M-file easier to read. To get the full syntax in the Model M-file, follow these steps:

- 1 From the **Options** menu, open the **Preferences** dialog box.
- 2 Click the **Modeling** tab.
- 3 Clear the **Shrink coefficients** check box in the **Model M-file/FEM export** area.
- 4 Click **OK**.

Resetting the Model and the Model M-file Command Sequence

The Model M-file command sequence can sometimes become quite long. The **Reset Model** command removes everything but the geometry-related commands from the Model M-file command sequence for the current model. It also removes all meshes and solutions in the model, so you might have to solve the model again. To reset the Model M-file and remove the meshes and solutions, choose **Reset Model** from the **File** menu.

Using the COMSOL Multiphysics Model Library

The COMSOL Multiphysics Model Library is a collection of solved models from different application areas. All the files can be loaded as Model MPH-files for inspection and modification. Most models are also available as Model M-files. The models in the COMSOL Multiphysics Model Library provide a convenient starting point for models that share similar properties but have different geometry, material properties, and boundary conditions.

You find the Model Library models in the **Model Navigator** by clicking the **Model Library** tab. Choose **Open Model Library** from the **File** menu to open the **Model Navigator** with **Model Library** tab active.

Browse the Model Library using the list to the left. The models are grouped into application areas. Each model contains a model description and a model image, which you can see in the right part of the **Model Navigator**. To view the documentation for a model, select the model from the Model Library list and click the **Documentation** button. To load a model into COMSOL Multiphysics, double-click the model in the list or select a model and then click **OK**. The program sets the path to the Model Library during installation. If you need to set the root directory for the Model Library, click the **Library Root** button. To access the documentation for the current model choose **Model Documentation** from the **Help** menu.

Note: The COMSOL Multiphysics modules provide their own model libraries that work in the same way as the COMSOL Multiphysics Model Library but with a focus on the application areas that the module covers.

Models in the Model Library

The COMSOL Multiphysics Model Library consists of about 80 fully solved models which are documented in the *Model Library*. See the “Model Library Guide” on page 5 in the *COMSOL Multiphysics Model Library* for a table with information about all available models and page references to the documentation for each model.

Creating a User Model Library

You can create your own user model libraries for easy access and documentation of COMSOL Multiphysics models. To browse the user models, click the **User Models** tab in the **Model Navigator**. Specify the path to the user model library by clicking the **Library Root** button. If you saved a model image along with a description they appear in the **Model Navigator** when you select the model. If there is HTML-based documentation for the model, click the **Documentation** button to view it.

Saving Models for a User Model Library

Save your models as Model MPH-files to dedicated folders to begin building your own model library. To add information about the model use the **Model Properties** dialog box and the **Save Model Image** command.

Glossary

This glossary contains terms related to finite element modeling, mathematics, geometry, and CAD as they relate to the COMSOL Multiphysics software and documentation. For more application-specific terms, see the glossaries in the documentation for the following modules: *AC/DC Module*, *Acoustics Module*, *CAD Import Module*, *Chemical Engineering Module*, *Earth Science Module*, *Heat Transfer Module*, *MEMS Module*, *RF Module*, and *Structural Mechanics Module*. For references to more information about a term, see the index.

Glossary of Terms

adaptive mesh refinement A method of improving solution accuracy by adapting the mesh to the problem's physical behavior.

affine transformations Geometric transformations that are combinations of linear transformations and translations.

algebraic multigrid (AMG) An *algebraic multigrid* solver or preconditioner performs one or more cycles of a multigrid method using a coarsening of the discretization based on the coefficient matrix. Compare to *geometric multigrid (GMG)*.

analysis type The *analysis type* is a top-level setting for the analysis of a model in most of the *physics modes*. The analysis type is available in the Model Navigator when starting a new model and in the **Application Mode Properties** and **Solver Parameters** dialog boxes. The analysis type determines suitable solver settings and, in some cases, equation settings. Typical analysis types include stationary, eigenfrequency, transient, time-dependent, and parametric analyses. For multiphysics models, there is typically one analysis type setting for each application mode.

analyzed geometry A supported geometry description for modeling and analysis. It can be of any of the formats *composite geometry object*, *assembly*, *Geometry M-file*, or *mesh*.

anisotropy Variation of material properties with direction.

application mode A predefined interface or template for a specific type of physics or equation. Each application mode provides its own set of boundary conditions, material properties, equations, and postprocessing variables.

application programming interface (API) The *API* provides a set of documented functions and methods for interacting with COMSOL Multiphysics. Using the API, users can create their own applications with customized geometries, equations, and so on.

application scalar variable A scalar variable with the current geometry as its context, for example, an angular frequency. Application scalar variables are valid in the all of the model's current geometry and include, but are not limited to, scalar numeric values.

arbitrary Lagrangian-Eulerian formulation (ALE formulation) A formulation for a moving mesh where dependent variables represent the mesh displacement or mesh velocity. The COMSOL Multiphysics solvers have built-in support for the mesh movement.

arc A segment of the circumference of a circle or ellipse.

Argyris element A 2D, 6-node triangular finite element with a 5th-order *basis function* providing continuous derivatives between elements.

aspect ratio The ratio between the longest and shortest element or geometry dimension.

assemble Taking the local element stiffnesses, masses, loads, and constraints to form the *stiffness matrix*, *mass matrix*, load vector, constraint matrix, and constraint residual vector.

assembly An *analyzed geometry* where the original geometry objects remain as individual parts.

associative geometry An algorithm that maps data associated with a geometry to the new geometry entities when the geometry is modified.

backward differentiation formula (BDF) A multistep formula based on numerical differentiation for solutions to *ordinary differential equations*. A BDF method of order n computes the solution using an n th-grade polynomial in terms of backward differences.

basis function A function φ_i in the *finite element space* such that the i th degree freedom is 1, while all other degrees of freedom are 0. For the Lagrange finite element space, φ_i is a linear or higher order polynomial on each mesh element with value 1 in node i and 0 in all other nodes.

Bernstein polynomial See *Bézier basis*.

Bézier basis A set of polynomial functions that occur in the definition of a *Bézier curve*. These polynomial functions are often called *Bernstein polynomials*.

Bézier curve A *rational Bézier curve* is a parameterized *curve* formed as the quotient of two polynomials expressed in the Bézier basis. It is a vector-valued function of one

variable. The coefficients of a rational Bézier curve are geometrically interpreted as *control points* and *control weights*. A *nonrational Bézier curve* is a rational Bézier curve with all weights equal, thereby making the denominator polynomial equal to a constant. A nonrational Bézier curve is also called an *integer Bézier curve*.

Bézier patch, Bézier surface A *Bézier patch* or *Bézier surface* is a surface extension of the *Bézier curve*. The *Bézier patch* is a function of two variables with an array of control points.

Boolean operations Boolean operations are used to construct a *geometry object* from other solid geometry objects and rebuild it in a new form. At least two primary geometry objects are required to create a resultant new geometry object. That new object depends on the type of Boolean operation:

- Union (add): the resultant geometry object occupies all the space of the initial geometry objects
- Difference (subtract): the resultant geometry object occupies all the space of the first geometry object except for the space inside the second geometry object.
- Intersection: the resultant geometry object occupies only the space common to the initial geometry objects

boundary A *domain* with a space dimension one less than the space dimension for the geometry, for example, a *face* in a 3D geometry. In a mathematical context, the symbol $\partial\Omega$ represents the boundary of the domain Ω . Sometimes *boundary* is used in a narrower sense meaning an *exterior boundary*. See also *interior boundary*, *exterior boundary*.

border The interface between two parts in an *assembly*.

boundary modeling A geometry modeling method to create a geometry by defining its boundaries. Compare to *solid modeling* and *surface modeling*.

brick element See *hexahedral element*.

chamfer A CAD feature that trims off a corner with a plane or straight line.

Cholesky factorization A memory-saving version of *LU factorization* where U is the transpose of L . It requires that the coefficient matrix A ($A = LU$) be a symmetric positive definite matrix. See also *LU factorization* and *positive definiteness*.

coefficient form PDE A PDE in the coefficient form is a PDE formulation suited for linear PDEs

$$\left\{ \begin{array}{ll} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f & \text{in } \Omega \\ \mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) + q u = g - \mathbf{h}^T \boldsymbol{\mu} & \text{on } \partial\Omega \\ h u = r & \text{on } \partial\Omega \end{array} \right.$$

coerce To convert a *geometry object* from one type to another, for example, from a *curve object* to a *solid object*.

composite geometry object, composite solid object Geometric objects made up by combining *primitive geometry objects* and other composite objects. See also *constructive solid geometry*, *primitive geometry object*, and *Boolean operations*.

COMSOL Multiphysics binary file A binary data file with the extension `.mphbin` that contains geometry objects or mesh objects. Earlier versions of COMSOL Multiphysics used the file extension `.flb`.

COMSOL Multiphysics text file A text data file with the extension `.mph.txt` that contains geometry objects or mesh objects.

condition number A measure of the possible error in a solution due to ill-conditioning of the equations. See also *ill-conditioning*.

constant A named model property that has a constant numeric value.

constraint Restriction imposed upon the dependent variables, typically as a *Dirichlet boundary condition*. *Neumann boundary conditions* are not regarded as constraints. When Dirichlet boundary conditions are introduced, the finite element algorithm makes a corresponding change to the Neumann boundary conditions so that the resulting model becomes solvable. For an *ideal constraint*, COMSOL Multiphysics accomplishes this change by adding the transpose of the constraint matrix \mathbf{h} times a vector of *Lagrange multipliers* to the right-hand side of the Neumann boundary condition. For a *non-ideal constraint*, the extra term is often some matrix times the vector of Lagrange multipliers. In a mechanical model, the extra term is called a *constraint force*.

constructive solid geometry (CSG) A solid-modeling method that combines simple solid shapes, or *primitives*, to build more complex models using Boolean operations. See also *solid modeling* and *primitive*.

control point *Bézier* and *NURBS* curves and surfaces are defined by a set of points known as *control points*. The locations of these points control the curve's shape.

control weight Scalar values assigned to *control points* to further control the shape of a curve or surface.

contour plot A plot that shows the variation of a solution component or other quantity. Points with equal values of the plotted quantity are connected with contour lines.

convergence The tendency for a finite element solution to approach the exact solution within well-defined and specified tolerances, for example, by reducing the mesh element size or the time step.

coupling variable A variable used to couple solution data within a domain or between different domains. See also *extrusion coupling variable*, *projection coupling variable*, and *integration coupling variable*.

curl element See *vector element*.

curve The path of a point moving through space. See also *Bézier curve*, *NURBS*, and *manifold*.

curve object A geometry object consisting of only *edges* and *vertices*, for example a geometry object representing a *curve*.

curve segment An individual polynomial or rational polynomial curve. Compounded curves consist of several *curve segments*.

degree of freedom (DOF) One of the unknowns in a discretized finite element model. A degree of freedom is defined by a name and a *node point*. The degree of freedom names often coincide with the names of the dependent variables. The local degrees of freedom are all degrees of freedom whose node points are in one mesh element.

dependent variable A varying quantity whose changes are arbitrary, but they are regarded as produced by changes in other variables. For example, temperature is a

function of the space coordinates and time. In a narrower sense, the dependent variables, or *solution components*, are the unknowns in a mathematical PDE model. Compare to *independent variable*.

differential-algebraic equation (DAE) A set of equations that includes both differential and algebraic equations. A DAE is classified in terms of its *index*, a positive integer, which is related to the minimum number of differentiations needed to transform a DAE to an ODE form.

direct solver A solver for a system of linear equation that uses some variant of Gaussian elimination. Compare to *iterative solver*.

Dirichlet boundary condition A Dirichlet boundary condition specifies the value of the function (dependent variable) on a boundary. Dirichlet boundary conditions are sometimes called *essential boundary conditions* or *constraints*. For a coefficient form PDE the Dirichlet boundary condition is

$$hu = r.$$

See also *constraint*.

discretization The process of dividing a continuous system into a finite number of elements with finite size. The difference between the finite-element representation and the real system, the discretization error, drops as the size of the elements decrease. For a time-dependent analysis, a discretization of time into steps provides an idealized behavior of the variations in the solution during these steps.

divergence element A finite element often used for electromagnetic vector fields. The degrees of freedom on the boundary of a mesh element correspond to normal components of the field. Also *Nédélec's divergence element*.

domain A topological entity within a geometry model that describes bounded parts of the manifolds in the model, and also the relations between different manifolds in the geometry. The different domain types are the *vertex*, *edge*, *face*, and *subdomain*. A domain of dimension one less than the space dimension is referred to as a *boundary*. See also *manifold*.

domain group Domains that use the same physical properties, coefficient values, or boundary conditions form a *domain group*. A data structure called an *index vector* connects the settings for each group to the corresponding domains.

drop tolerance A nonnegative scalar used in the incomplete LU preconditioner for the iterative solvers. See *incomplete LU factorization*.

dynamic model See *time-dependent model*.

edge, edge segment A domain representing a bounded part of a *curve*. An *edge* or *edge segment* is a *boundary* in a 2D geometry. See also *domain*.

edge element See *vector element*.

eigenvalue PDE A PDE that describes an eigenvalue problem with unknown eigenmodes (eigenfunctions) u and eigenvalues λ . The *coefficient form* eigenvalue PDE is:

$$\lambda^2 e_a u - \lambda d_a u + \nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + a u = 0$$

elliptic PDE A linear stationary 2nd-order elliptic PDE has the form

$$\nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

where c is positive or negative definite, for example, Poisson's equation.

embed To insert a 2D geometry into a 3D geometry model.

equation system form The form of the PDE that the **Equation System** dialog boxes use to display the system of equations. See also *coefficient form*, *general form*, *solution form*, and *weak form*.

error Deviations from the correct solution, primarily due to: poor modeling; *discretization* (such as insufficiently fine mesh, poor elements, or insufficiently short time steps); and roundoff and truncation (depending on numerical representation, *ill-conditioning*, or the solution algorithms).

error estimate An estimation of the error in the numeric solution to a problem, either locally or globally, primarily for use by an adaptive mesh refinement. See also *adaptive mesh refinement*, *error*.

equivalent boundaries *Boundaries* that are rigid transformations of each other and have compatible meshes. See also *periodic boundary condition*.

essential boundary condition See *Dirichlet boundary condition*.

expression variable A user-defined variable that is defined on any geometry domain in terms of *dependent variables*, *independent variables*, *constants*, *application scalar variables*, and other *expression variables*. Global expression variables are valid in all geometries; scalar expression variables are valid in the current geometry.

extended mesh A data structure that includes the full finite element mesh. See also *mesh*, *node point*.

extended multiphysics A model that includes nonlocal couplings and dependencies between variables, where the value at a point is the result of a computation elsewhere in the domain or in another geometry defined in the same model. *Coupling variables* provide the ability to project or extrude values from one geometry or domain to another. Compare to *multiphysics*.

exterior boundary An *exterior boundary* for a dependent variable u is a *boundary* such that u is defined only on one of the adjacent subdomains, that is, a boundary to the exterior of the computational domain. See also *boundary*.

extrude To create a 3D geometry object from a 2D geometry object in a *work plane* by translating (extruding) it along a path, often a straight line.

extruded mesh A 3D mesh created by extrusion of a 2D mesh. An extruded mesh can contain *hexahedral elements* and *prism elements*.

extrusion coupling variable A variable in the destination domain that takes values from the source domain by interpolation at points that depend on the position of the evaluation points in the destination domain.

face, face segment A domain describing a bounded part of a *surface* in a 3D geometry. A *face* or *face segment* is a *boundary* in a 3D geometry. See also *domain*.

face object A geometry object with no topological information on subdomains. Typically a trimmed surface is represented as a face object.

FEM See *finite element method*.

FEM structure The main data structure, containing all data for a model.

fillet A curved transition from one boundary to another, creating a rounded corner.

finite element In the mathematical sense, a *mesh element* together with a set of *shape functions* and corresponding *degrees of freedom*. The linear combinations of the shape functions form a space of functions called the *finite element space*. In the traditional FEA sense, the concept of a finite element also includes the discretized form of the PDEs that govern the physics. COMSOL Multiphysics generally uses *finite element* in the mathematical sense.

finite element analysis (FEA) A computer-based analysis method for field problems using the *finite element method*.

finite element method (FEM) A computational method that subdivides an object into very small but finite-size elements. The physics of one element is approximately described by a finite number of *degrees of freedom (DOFs)*. Each element is assigned a set of characteristic equations (describing physical properties, boundary conditions, and imposed forces), which are then solved as a set of simultaneous equations to predict the object's behavior.

finite element space The linear space of functions where the finite element approximation to the solution of a PDE problem is sought. The functions in the finite element space are linear combinations of *basis functions (shape functions)*.

flux vector The vector $\Gamma = -c\nabla u - \alpha u + \gamma$. See also *generalized Neumann boundary condition* and *normal flux*.

free mesh An *unstructured mesh* that can represent any geometry. Compare to *mapped mesh*.

free mesher The mesh generator creating *free meshes*. The mesh generator creating *triangular elements* is also referred to as the *free triangle mesher*, and the mesh generator creating *quadrilateral elements* is also referred to as the *free quad mesher*.

free quad mesher The mesh generator creating unstructured quadrilateral meshes.

free triangle mesher The mesh generator creating unstructured triangular meshes.

Gauss point Sometimes improperly used as a synonym for *integration point*. See also *integration point*.

general form PDE A PDE in the general form is a PDE formulation suited for nonlinear PDEs

$$\begin{cases} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot \Gamma = F & \text{in } \Omega \\ -\mathbf{n} \cdot \Gamma = G + \left(\frac{\partial R}{\partial u} \right)^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

generalized Neumann boundary condition A generalized Neumann boundary condition (also called a *mixed boundary condition* or a *Robin boundary condition*) specifies the value of a linear combination of the *normal flux* and the *dependent variables* on a boundary. For a coefficient form PDE, the generalized Neumann boundary condition is

$$\mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) + qu = g - h^T \mu$$

The generalized Neumann condition is often called just *Neumann condition* in the documentation.

geometric multigrid (GMG) A *geometric multigrid* solver or preconditioner performs one or more cycles of a multigrid method, using a coarsening of the discretization based on a coarsening of the mesh or a reduction in the order of the shape functions. Compare to *algebraic multigrid (AMG)*.

Geometry M-file A COMSOL Script M-file containing a 1D or 2D geometry description using *vertices* and *intervals* (1D) or parameterized *edge segments* (2D).

geometry model A collection of topological and geometric entities that form a complete geometric description of the model. The geometric entities that make up the geometry model are also called *manifolds*, and the topological entities are referred to as *domains*.

geometry object The objects that represent a geometry model. See also *curve object*, *face object*, *primitive geometry object*, *solid object*.

grid A *grid* usually refers to sets of evenly-spaced parallel lines at particular angles to each other in a plane, or the intersections of such lines. Compare to *mesh*.

Hermite element A finite element similar to the *Lagrange element*. The difference is that there are degrees of freedom for the (1st-order) space derivatives at the mesh vertices. See also *Lagrange element*.

hexahedral element A 3D mesh element with eight corners and six faces, also referred to as *brick element*; sometimes also called *hex element* as a short form.

higher-order element A finite element with *basis functions* that consists of polynomials of degree 2 or higher.

hybrid geometry modeling Creating a geometry model using a combination of *boundary modeling/surface modeling* and *solid modeling*.

hyperbolic PDE A typical example of a linear 2nd-order hyperbolic PDEs is the *wave equation*

$$e_a \frac{\partial^2 u}{\partial t^2} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

where e_a and c are positive.

ideal constraint See *constraint*.

identity condition A special case of the *extrusion coupling variable* used to couple solutions between identical coordinate points in different geometries.

identity pair A *pair* that connects parts in an *assembly* using constraints on the equations so that the solution becomes continuous across the border between the parts.

IGES file An *IGES file* contains 3D CAD data, including the 3D geometry, in an open format according to the *Initial Graphics Exchange Specification*. You can import an IGES file to COMSOL Multiphysics using the CAD Import Module.

ill-conditioning An ill-conditioned system is sensitive to small changes in the inputs and is susceptible to roundoff errors. See also *condition number*.

imprint An imprint of the smaller boundary on the larger boundary that makes the parts in a *pair* match. An imprint inserts points on the boundary in 2D and creates a curve on the boundary in 3D.

incomplete LU factorization An approximate *LU factorization* where small matrix elements are discarded to save memory and computation time. The *drop tolerance* is a relative measure of the smallness of the elements that should be discarded. See also *LU factorization*.

independent variable A variable that can cause variation in a second, *dependent variable*. The independent variables are most often space coordinates and time. Compare to *dependent variable*.

index, for DAE See *differential-algebraic equation*.

index vector See *domain group*.

initial condition The starting values for the dependent variables in a time-dependent analysis and for *nonlinear iterations* or other iterative solvers.

integration coupling variable A variable that evaluates integrals of expressions over the domains in the source and returns a single scalar value available in the destination domain.

integration point See *numerical integration formula*.

interactive meshing Building a mesh in an incremental fashion where each meshing operation acts on a set of geometry domains.

interior boundary An *interior boundary* for a dependent variable u is a *boundary* such that u is defined on both adjacent subdomains or in no adjacent subdomain. See also *boundary*.

interval The domain between two vertices in a 1D geometry. Also called a *subdomain*.

isoparametric element A finite element that uses the same *shape function* for the element shape coordinates as for the *dependent variables*.

isosceles triangle A triangle with at least two equal sides (and two equal angles).

iteration See *iterative solver*.

iterative solver A solver for a system of linear equations that uses an iterative method, computing a sequence of more and more accurate approximations to the solution.

Each step in this sequence is one *linear iteration*. This should not be confused with the Newtons iterations (*nonlinear iterations*) that occur in the solution of a nonlinear system of equations. Compare to *direct solver* and *nonlinear iteration*.

Jacobian matrix A matrix containing the first derivative of a vector-valued function of a vector variable. In particular, it is the derivative of the *residual vector* with respect to the *solution vector*. When used in this narrower sense, the term *stiffness matrix* is sometimes used.

Lagrange element A *finite element* with polynomial shape functions of a certain *order*. The value of the function is used as the *degree of freedom*, and the node points are evenly distributed within the mesh element.

Lagrange multiplier An extra dependent variable introduced in the Neumann boundary condition when a constraint is added. See also *constraint*.

linear iteration A step in a linear iterative solver. See *iterative solver*. Compare to *nonlinear iteration*.

linear PDE An equation where both sides are sums of a known function, the unknown functions, and their partial derivatives, multiplied by known coefficients that only depend on the *independent variables*. Other PDEs are called *nonlinear*.

loft To create a 3D geometry by smoothly fitting a surface to a series of cross sections.

LU factorization For a linear system of equations, a version of Gaussian elimination that produces a factorization $A = LU$ of the coefficient matrix, where L and U are the lower and upper triangular matrices, respectively. This makes it easy to quickly solve a number of systems with the same coefficient matrix. See also *direct solver*.

manifold A mathematical function describing a *surface*, *curve*, or *point* in a geometry model of any dimension. See also *domain*.

mapped mesh A *structured mesh* with *quadrilateral elements* generated by mapping using transfinite interpolation.

mapped mesher The mesh generator creating *mapped meshes*.

mass matrix The matrix E that multiplies the second time derivative of the *solution vector* in the linearized discretized form of a PDE problem. If there are no second time

derivatives (i.e., if $E=0$), then the term mass matrix is often used for the matrix D that multiplies the first derivative of the solution vector (the D matrix is otherwise called the *damping matrix*).

mesh A subdivision of the domains of a geometric model into, for example, triangles (2D) or tetrahedra (3D). These are examples of *mesh elements*. See also *grid*, *structured mesh*, and *unstructured mesh*.

mesh element The individual elements in the mesh that together form a partitioning of the geometry, for example, *triangular elements* and *tetrahedral elements*. See also *finite element*.

mesh vertex An endpoint or corner of a mesh element. See also *node point* and *vertex*.

method of lines A method for solving a time-dependent PDE through a space discretization, resulting in a set of ODEs.

mixed boundary condition See *generalized Neumann boundary condition*.

mode reduction A model-reduction technique for reducing systems with many degrees of freedom, for example large finite element models, to a form with fewer degrees of freedom for dynamic system simulations and analysis. See also *state-space model*.

Model MPH-file A binary data file with the extension `.mph` that contains a COMSOL Multiphysics model. Often also just called model file. Earlier versions of COMSOL Multiphysics used the file extension `.fl`.

Model M-file An M-file containing commands that create a COMSOL Multiphysics model. Model M-files are text files that can be modified and used both with COMSOL Script and with MATLAB. The COMSOL Multiphysics graphical user interface can load a Model M-file. Compare to *Model MPH-file*.

MRI data *Magnet resonance imaging (MRI) data* is an image data format, primarily for medical use. MRI produces high-quality images of the inside of the human body. 3D MRI data is usually represented as a sequence of 2D images.

multidisciplinary models Multidisciplinary models combine PDE-based finite element modeling with other mathematical modeling techniques such as dynamic simulation in areas like automatic control and signal processing.

multigrid A solver or preconditioner for a linear system of equations that computes a sequence of increasingly accurate approximations of the solution by using a hierarchy of coarsened versions of the linear system (having fewer degrees of freedom). See also *algebraic multigrid*, *geometric multigrid*.

multiphysics Multiphysics models include more than one equation and variable from different types of physics. These variables can be defined in different subdomains. The equations can be coupled together through equation coefficients that depend on variables from other equations. Compare to *extended multiphysics*.

natural boundary condition See *Neumann boundary condition*.

Neumann boundary condition A Neumann boundary condition specifies the value of the *normal flux* across a boundary. Neumann boundary conditions are sometimes called *natural boundary conditions*. Compare to *generalized Neumann conditions*.

Newton's method An iterative solver method, also called the *Newton-Raphson method*, for solving nonlinear equations. See also *nonlinear iterations*.

Newton-Raphson method See *Newton's method*.

node point Any point in the mesh element where the degrees of freedom are defined. The node points often include the mesh vertices and possible interior or midpoint locations. See also *degree of freedom* (DOF) and *mesh vertex*.

non-ideal constraint See *constraint*.

nonlinear iteration A Newton step in the solution of a nonlinear PDE problem. Each nonlinear iteration involves the solution of a linear system of equations. Compare to *linear iteration*.

nonlinear PDE See *linear PDE*.

norm A scalar measure of the magnitude of a vector or a matrix. Several types of norms are used to measure the accuracy of numerical solutions.

numerical integration formula A numeric-integration method that approximates an integral by taking the weighted sum of the integrand evaluated at a finite number of points, the *integration points* (sometimes improperly called *Gauss points*). Also called *quadrature formula*.

normal flux The normal component of the *flux vector* at a boundary.

NURBS The *nonuniform rational B-spline (NURBS)* is a popular curve and surface representation scheme. A NURBS representation can be divided into a number of *rational Bézier curves* or surfaces.

order of a finite element The degree of the polynomials that define the *shape functions (basis functions)*.

ordinary differential equation (ODE) An equation involving functions and their derivatives. The derivatives are with respect to one independent variable only. Compare to *partial differential equation (PDE)*.

pair Two sets of domains (for example, boundaries): one set with domains from one part, and a second set with domains in another part of an *assembly*.

parabolic PDE A typical example of a linear 2nd-order parabolic PDE is the *heat equation*

$$d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f$$

where d_a and c are positive.

parameter A constant that takes on different values for each model in a parametric analysis. See also *constant*.

partial differential equation (PDE) An equation involving functions and their partial derivatives; that is, an equation that includes derivatives with respect to more than one independent variable. Compare to *ordinary differential equation (ODE)*.

PDE mode A type of *application mode* for modeling using PDEs. The PDE modes provide direct access to general PDE coefficients and boundary conditions. Compare to *physics modes*.

periodic boundary condition A boundary condition where the values of the solution appear in a periodic pattern, typically so that the value of the solution on one boundary is equal to the value on another boundary. See also *equivalent boundaries*.

phasor A complex number or a vector of complex numbers representing a sinusoidally varying current or voltage.

physics modes *Application modes* for different types of physics in the COMSOL Multiphysics graphical user interface. These application modes contain predefined equations and boundary conditions. Compare to *PDE modes*.

pivot Usually a value on the main diagonal of the *stiffness matrix*. *Pivoting* is the interchanging of rows and columns in order to place a particularly large element in the diagonal position. The value of the diagonal term when it is used to eliminate values below it is called the *pivot value*.

point A location in space.

point object A geometry object with only *vertices*.

positive definiteness A symmetric matrix is *positive definite* when all its eigenvalues are positive.

post data Data format used by the COMSOL Multiphysics postprocessing and visualization routines.

preconditioner The convergence rate of iterative methods depends on the spectral properties of the coefficient matrix. A *preconditioner* is a matrix that transforms the linear system into one that has the same solution but that has more favorable spectral properties. See also *algebraic multigrid*, *geometric multigrid*, *incomplete LU factorization*, *iterative solver*, and *SSOR*.

primitive, primitive geometry object A geometry object with a basic shape such as a cube or a sphere. You can add primitives to a model, using arbitrary sizes and positions, and combine them to form complex shapes. See also *constructive solid geometry*, *composite geometry object*, and *Boolean operations*.

prism element A 3D mesh element with six corners and five faces, also referred to as *wedge element*.

projection coupling variable A variable that takes values from the source domain by evaluating line integrals over lines whose positions are dependent on the position of the evaluation points in the destination domain.

quadrature formula See *numerical integration formula*.

quadrilateral element A 2D mesh element with four corners and four edges; sometimes also called *quad element* as a short form.

rational Bézier curve See *Bézier curve*.

residual vector The vector L in the discretized form of a PDE problem. In the absence of *constraints*, the discrete form of a stationary equation is $0 = L(U)$ where U is the *solution vector*.

revolve To create a 3D geometry object from a 2D geometry object in a *work plane* by rotating it around an axis.

revolved mesh A 3D mesh created by revolving a 2D mesh. A revolved mesh can contain *hexahedral elements* and *prism elements*.

Robin boundary condition See *generalized Neumann boundary condition*.

ruling application mode The *application mode* in a *multiphysics* model that determines the *analysis type*. It is possible to select the ruling application mode in the Model Navigator.

shape function A *basis function* described in local element coordinates. See also *basis function*.

shift A value σ around which an eigensolver searches for eigenvalues.

simplex element *Triangle element* in 2D and *tetrahedral element* in 3D.

Simulink structure A MATLAB data structure for use in a Simulink model. The *Simulink structure* contains the COMSOL Multiphysics subsystem, including inputs and outputs.

solid A description of a part of the modeling space. See also *subdomain*.

solid modeling A 3D geometry modeling method that describes both the boundary and interior of the geometry using solid objects. See also *constructive solid geometry (CSG)* and *solid*.

solid object A geometry object representing one or several *solids*.

solution component See *dependent variable*.

solution form The form of a PDE that the solver uses. See also *coefficient form*, *equation system form*, *general form*, and *weak form*.

solution matrix A matrix that contains a sequence of solutions as columns. A steady-state problem results in a *solution vector*, but eigenvalue problems, time-dependent problems, and parametric analyses produce a *solution matrix*. See also *solution structure*.

solution structure A data structure that includes the *solution vector* or *solution matrix* and any associated data such as parameter values, output times, or eigenvalues.

solution vector A vector with all the *degrees of freedom* (values of the *dependent variables*) as its components. See also *solution matrix* and *solution structure*.

solver scripting To record and replay a sequence of solver commands and other related commands using a scripting language.

sparse matrix Matrix for which the number of zero elements is large enough to justify special data types and algorithms that avoid operations on zero elements.

split To divide a geometry object into its minimal parts.

stability A solver for a time-dependent model is *unconditionally stable* if the initial conditions are not amplified artificially and the roundoff errors do not grow, regardless of the size of the time step. A solver is *conditionally stable* if there is a maximum value of the time step above which the numerical solution is unstable.

state-space model A linear time-invariant representation of a dynamic system as a set of 1st-order *ODEs* of the form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ \dot{y} &= Cx + Du\end{aligned}$$

where x is the state vector, u is the input, and y is the output. A , B , C , and D are the constant dynamics, input, output, and direct transmission matrices, respectively.

static model See *stationary model*.

stationary model A model where the dependent variables do not change over time. It typically represents a steady-state solution. Also called *static model* or *steady model*.

steady model See *stationary model*.

stiffness matrix See *Jacobian matrix*.

streakline The locus of particles that have earlier passed through a prescribed point in space. See also *streamline*.

streamline A curve that is everywhere tangent to the vector field (in particular a velocity field) at a given instant of time. Sometimes called a *flow line* or *flux line*. See also *streakline*.

streamline-diffusion stabilization A numerical technique for stabilization of the numeric solution to a PDE by artificially adding diffusion in the direction of the *streamlines*.

strong form A partial differential equation in the *strong form* is the standard formulation as an equality of functions. The strong form is divided into the *coefficient form* and the *general form*. Compare to *coefficient form*, *general form*, and *weak form*.

structured mesh A mesh for which all elements and nodes have the same topology. Compare to *unstructured mesh*.

subdomain A topological part of the modeling space in a geometry model. The geometric representation of a subdomain is a line segment (interval) in 1D, an area in 2D, and a volume in 3D. In a mathematical context, the symbol Ω represents the domain where the equations are defined. See also *domain*.

surface A mathematical function (manifold) from 2D to 3D space.

surface normal A vector perpendicular to the surface.

surface modeling A 3D geometry modeling method to describe a geometry by defining its bounding surfaces. Compare *boundary modeling* and *solid modeling*.

swept mesh A 3D mesh generated by sweeping a face mesh along a subdomain.

symmetric matrix A matrix that equals its own transpose.

symmetric successive overrelaxation (SSOR) A *symmetric successive overrelaxation (SSOR)* preconditioner uses classic SSOR iterations.

symmetry The invariance of an object attribute or of the object itself under a well-defined operation or transformation such as inversion, rotation, or reflection. A *symmetry* allows for a reduction of the model geometry so that appropriate boundary conditions account for the redundant portions of the geometry. Axisymmetry is a common type of *symmetry*.

symmetry boundaries See *equivalent boundaries*.

test function See *weak form*.

tetrahedral element A 3D mesh element with four corners, six edges, and four triangular faces.

time-dependent model A model where at least one of the dependent variables changes over time, for example, the heat equation or the wave equation. Also called *dynamic model*, *transient model*, or *unsteady model*.

transient model See *time-dependent model*.

triangular element A 2D mesh element with three corners and three edges.

trimmed surface If the parameter space of a surface is divided into “valid” and “invalid” regions, the image of the valid regions is called the *trimmed surface*. This corresponds to the part of the surface limited by a closed loop of edges lying on the surface.

unstructured mesh A mesh without a specific pattern where the elements can have different shapes and the nodes can have different connectivities. Compare to *structured mesh*.

unsteady model See *time-dependent model*.

vector element A finite element often used for electromagnetic vector fields. Each mesh element has degrees of freedom corresponding only to tangential components of the field. Also called *curl element*, *Nédélec's edge element*, or just *edge element*.

vertex A point in a geometry model, often an endpoint of a geometry segment or an intersection of geometry entities of higher degree such as *edges* or *faces*. A vertex is referred to as a *point* for the specification of point sources and other PDE modeling. See also *domain*.

wave extension An extension of a time-dependent PDE or system of PDEs into a corresponding system using 2nd-order derivatives with respect to time, which is typical for wave equations. The wave extension adds equations that define the 1st-order time derivatives of the original dependent variables.

weak constraint A reformulation of a *Dirichlet boundary condition* as a *weak form* equation. When using a weak constraint, the corresponding *Lagrange multiplier* becomes a *solution component (dependent variable)*.

weak form A partial differential equation in the *weak form* is a more general formulation than the strong form. It is produced by multiplying the *strong form* PDE with an arbitrary function called the *test function* and integrating over the computational domain. Compare to *strong form*.

wedge element See *prism element*.

work plane An embedded 2D work space that can be positioned relative to the coordinate planes or an already existing geometry. Using *work planes* makes it possible to define a geometry in terms of previously created geometry objects such as *points*, *edges*, and *faces*.

I N D E X

- 3D geometry objects
 - visualization of 98
- 3D primitives, editing 75
- A**
 - A4 paper format 557
 - absolute IPS system 189
 - absolute tolerance 392
 - AC/DC Module
 - application areas 12
 - application modes 12
 - material properties supported in 12
 - accessing solutions 166
 - accurate derivative recovery 167, 469
 - accurate integrals of reaction forces 168
 - Acoustics Module
 - application areas 13
 - application modes 13
 - adaptive mesh refinement 419
 - adaptive solver
 - termination criteria for 421
 - using 420
 - adj operator 168
 - adjacency selection methods 134
 - choosing 134
 - selecting objects using 134
 - adjacent objects, selecting all 134
 - adjoint sensitivity method 423
 - adjoint sensitivity operator 168
 - advancing front method 317
 - affine transformations 33
 - algebraic variables 401
 - alpha-blending value, for transparency
 - 112
 - ambient light 110
 - analysis types, selecting 378
 - analytic functions 150
 - animated GIF movie format 479
 - animations 478
 - as MATLAB movies 480
 - encoding format 480
 - anisotropic mapped mesh 328
 - anisotropic materials, modeling 221
 - anisotropic meshes 307
 - anisotropic properties 213
 - antialiasing 120, 494
 - application areas 5
 - application mode properties 379
 - application mode variables 176, 180
 - viewing and modifying 220
 - application modes
 - in the AC/DC Module 12
 - in the Acoustics Module 13
 - in the Chemical Engineering Module 15
 - in the Earth Science Module 16
 - in the Heat Transfer Module 16
 - in the MEMS Module 17
 - in the RF Module 18
 - in the Structural Mechanics Module 19
 - application scalar variables
 - specifying 265
 - arc length
 - on x-axis in plots 546
 - parameter for 171
 - arcs 47
 - area moment of inertia 79
 - areas, displaying 77
 - arrow keys, for orbit, pan, zoom operations 102
 - arrow plots 508
 - arrows, showing direction on boundaries
 - 171
 - aspect ratio for equal axis length 31
 - assembly 92

- associative geometry 94
- at operator 167
- augmented Lagrangian solver 390
- author name 571
- automatic solution form 382
- AVI movie format 479
- axes
 - controlling in 3D 31
 - equal increments for 25
 - settings for 25
- axisymmetric models
 - mapping to 3D 284
 - postprocessing in 3D 496
- B**
 - backward differentiation formula 395
 - backward differentiation formulas 398
 - BDF order 398
 - beam cross-section library 19
 - Bessel functions 148, 161
 - Bézier curves 46, 48
 - BiCGStab iterative solver 427
 - binary operators 147
 - bitmap graphics 493
 - bold font, indicating changed values 563
 - Boolean operations 24, 31
 - in 3D 67
 - on solid objects 42
 - set difference 32
 - set intersection 32
 - set union 32
 - boundary 35
 - exterior 246
 - interior 43, 246
 - boundary conditions
 - for identity pairs 253
 - for periodic boundaries 257
 - setting 247
 - boundary coupled equation variables
 - 176, 186
 - boundary coupled shape variables 180, 186
 - boundary distance variables 295
 - boundary elements 300
 - boundary integration 456, 550
 - boundary layer meshes 338, 359
 - boundary modeling 46
 - in 2D 46
 - boundary plots 507, 540
 - boundary representation of 3D solids 67
 - boundary segment, removing 93
 - boundary settings 564
 - modifying 249
 - bounding box, representing the geometry when moving it 101
 - brick elements 301
 - brick meshes, creating 328
 - British engineering units 189
 - British thermal unit 200
 - C**
 - CAD files 85
 - calorie 198
 - camera history 103, 552
 - camera movement, constraining 103
 - camera panning 102
 - camera projection
 - changing 103
 - orthographic 103
 - perspective 104
 - Camera toolbar 101
 - canceling solvers 447
 - Cartesian coordinate system, specifying
 - 27
 - center of mass 77
 - centroid 79
 - CGS units 188
 - chamfers, creating 49
 - changed values, indication of 563
 - Chemical Engineering Module

- application areas 14
- application modes 15
- circle 48
- circular arcs 47
 - creating 47
- codec 480
- coefficient form 380
- coefficients, locked 211, 249
- coercing
 - geometry objects 34
 - to a curve object 34
 - to a face object 34
 - to a solid object 34
- color range 505
- color scale 505, 535, 537
 - displaying 478
- coloring of domain groups 208
- colormaps
 - creating 471
 - list of 470
 - selecting 470
- command sequence in Model M-files 575
- complex-valued problems 383
- complex-valued quantities, visualization
 - of 457, 464
- composite geometry 92
- composite geometry objects 46
 - creating 31
- composite solid object 24
- COMSOL figure windows, plotting in
 - 465
- COMSOL Multiphysics files 364
 - importing 86
- COMSOL Multiphysics functions 2
- COMSOL Multiphysics models
 - opening 568
 - saving 568
- COMSOL Multiphysics, application areas
 - for 5
- COMSOL Script
 - exporting data to 492
 - using interpolation data from 160
- COMSOL Script functions, using in
 - COMSOL Multiphysics 161
- COMSOL web sites 21
- comsol.opt.s, changing the maximum
 - Java heap size 124
- conditional expressions 166
- confirming selections 130
- conflicting selections 249
- conic sections 48
- connected manifolds 35
- consistent initialization of DAE system
 - 400
- constant predictor 409
- constants
 - adding 139
 - adding descriptions of 140
 - deleting 140
 - mathematical 147
 - saving and loading 140
- constraint type
 - ideal 217
 - non-ideal 217
- constraints
 - edge 256
 - ideal and non-ideal 217
 - point 256
- continuation method 406
- contour labels 505
- contour levels 505
- contour plots 503
 - filled 505
- control polygon 47
 - for Bézier curves 48
- control vertices 47

- for Bézier curves 48
- convergence
 - for models with discontinuous functions 149
 - plots of 448
- convergence rate, during Newton iterations 397
- converting meshes 345
- convolution integral 167
- convolution integrals 165
- coordinate system display 31
- copy of plots 551
- copying
 - boundary meshes 360
 - geometry objects 34
 - physics settings 144
- coupling variables 176, 269
 - destination 269
 - effect on the Jacobian matrix 294
 - extrusion 275
 - integration 269
 - projection 284
 - scalar 269
 - solution form when using 294
 - source 269
- cross-section plots 544
 - in 2D points 527
 - line/extrusion in 2D 526
 - using 473
- curl-curl operators, preconditioners for 431
- current directory 161
- current solution, accessing 436
- cursor position, display of 31
- curvature factor, for mesh elements 303, 311
- curve object 46
- curve parameter 171
- curve segments, modifying shape and position of 55
- custom colormaps 471
- cylindrical coordinate system, specifying 27

D

- DAE 401
 - index of 400
- damped Newton method 390
- damping factor 397
- data
 - displaying in full precision 459
 - exporting to COMSOL Script 492
 - exporting to file 487
 - exporting to MATLAB 492
- default values, indication of 563
- deformed mesh, creating geometry from 369
- deformed shape plots 524, 542
- degeneracies in the geometry 94
- degree of rational Bézier curve 48
- Delaunay method 317
- deleting
 - boundary segment 93
 - interior boundaries 35
- dependent variable 177
- depends operator 165
- derivatives
 - operators for 164
 - tangential 179
- descriptions, adding to constants 140
- deselecting
 - all objects 126
 - selected geometry objects 132
- dest operator 165
- destination transformation 278
- determinant, of the Jacobian matrix 174
- differential variables 401
- differential-algebraic system 401

- differentiate button
 - boundary settings 250
 - subdomain settings 219
- differentiation operators 164
- direct printing 557
- direct solvers 427
- direction arrows, on 2D boundaries 171
- directional light 107
- Dirichlet boundary condition 247
- discontinuous functions 149
- discretization 388
- discretization error, in gradients 167
- discretized linearized model 388
- display of objects, suppressing 122
- distance between points 77
- distance to boundaries 295
- distributed parameter sweeps
 - for the parametric solver 406
 - from the Parametric Sweep dialog box
 - 383
- documentation 2
- dolly in/out 102, 551
- domain groups 206
 - creating 207
 - default group 207
 - deleting unused 207
 - line styles and colors for 208
- domain number, variable for 174
- domain plots
 - in 3D 548
 - line/extrusion in 2D 530
 - on surfaces in 2D 528
- domains
 - adding to a solid 45
 - in the geometry 35
 - names of 35
- down operator 165
- Draw mode, entering 36
- Draw toolbar 39, 56
- drawing multiple objects 31
- dvol variable 174
- DXF files
 - format 87
 - importing 87
- dynamic problems 391
- E** Earth Science Module
 - application areas 15
 - application modes 16
 - edge constraints 256
 - edge elements 300
 - edge integration 456
 - edge loads 256
 - edge plots 541
 - edge segments 43
 - edge settings 564
 - edge vertex distribution 306
 - edges 35
 - editing
 - 3D primitives 75
 - geometry objects 54
 - plots 552
 - reports 559
 - eigenfrequency model 261
 - eigenfrequency problems 402
 - eigenmodes, weights for prioritizing 422
 - eigenvalue problems 402
 - normalization of solutions to 403
 - eigenvalue shift 402, 434
 - eigenvalue solver 402
 - eigenvalue variable 181
 - eigenvalues 380
 - electromagnetic units 188
 - electronvolt 198
 - electrostatic units 188
 - element area ratio 343
 - element growth rate 303, 312

- element layers 329
- element refinement
 - for postprocessing 464
 - selection methods for 422
- element selection for postprocessing 465
- element size
 - maximum 309
 - variable for 174
- element surface normals 173
- element types
 - in the Structural Mechanics Module 18
 - specifying 216
- element volume ratio 343
- elements, curved vs. straight 174
- ellipse 48
- elliptic PDEs 434
- elliptic problems 434
- embedding a 2D geometry in 3D 66
- encoding format 480
- energy balances 15
- equation system form 186
- equation variables 176, 185
- equations
 - deactivating in subdomains 214
 - displaying and hiding 211
- error estimates, norms for 421
- exponential polynomial functions 152
- exporting
 - constants 140
 - data to files 487
 - mesh 364
 - scalar expression variables 142
- expression variables 141, 176, 181
- expressions
 - enter during postprocessing 454
 - used in postprocessing 141
 - variables for 141
- external data, using in models 236
- extruded mesh 301
- extruding
 - a 2D geometry into 3D 64
 - meshes 328
- extrusion coupling variables
 - applications using 283
 - general transformation of 277
 - linear transformations for 275
- extrusion sections, creating 64

F

- face segments 67
- faces 35
 - deleting 76
- Fahrenheit 195
- FEM. See finite element method
- field variables 176
- figure windows, toolbar in 551
- file formats
 - DXF files 87
 - model files 568
 - Model M-files 568
 - NASTRAN bulk data 364
 - STL files 89
 - VRML files 89
- file names of generated files 558
- filled contour plots 505
- fillets, creating 49
- finite element method 5
- floating contacts 274
- flux 266
- fluxes, accurate computation of 266
- foot-pound-second unit system 189
- forward sensitivity method 423
- forward sensitivity operator 168
- frames
 - association with 178
 - selecting for postprocessing 455
- free quadrilateral meshes 318
- fsens operator 168

- full numeric precision, displaying data using 459
- function search path 161
- functional sensitivity operator 168
- functional-based norm, for error estimate 421
- functions
 - analytic 150
 - based on interpolated data 154
 - input and output formats 161
 - making independent of unit system 193
 - mathematical 147
 - piecewise analytic 151
- G** Gaussian elimination 428
- general transformation 277
- generalized Neumann boundary conditions 247
- generalized-alpha integration method 395
- geometric multigrid preconditioner 373
- geometric parametric sweeps 384
- geometric primitives 24
 - in 2D 41
 - in 3D 58
- geometric properties 77
- geometric variables 170
- geometries
 - creating from meshes 369
 - including in plots 463
 - projecting in work plane 63
 - viewing 36
 - viewing simultaneously 124
- geometry edges 463
- geometry modeling concepts 24
- geometry objects
 - arraying 34
 - coercing 34
 - composite 46
 - copying and pasting 34
 - creating an array of 34
 - editing 54
 - mirroring 33
 - moving 33
 - rotating 33
 - scaling 33
 - specifying using dialog box 31
 - splitting 35
 - visualization of 98
- geometry repair 33
- geometry requirements
 - for mapped meshes 321
 - for swept meshes 333
- geometry section 560
- geometry shape order 187
 - using first order 374
- geometry types
 - assembly 92
 - composite 92
- GIF movie format 479
- global expressions 142
- gradients, components of 178
- gram calorie 198
- gravitational IPS system 189
- gray colormap 471
- grayprint colormap 471
- Greek characters 467
- Gregorian year 198
- grid
 - settings for 25, 27
 - snapping to 31
- grid lines 31
- group names
 - changing 207
 - displaying 207
- group selection 127
- groups

- line styles and colors for 208
 - of material properties 234
- H**
 - headlights 105, 552
 - Heat Transfer Module
 - application areas 16
 - application modes 16
 - Heaviside functions 149
 - Hermitian matrices 431
 - hexahedral elements 301
 - hexahedral mesh, creating 328
 - highlighting
 - saving preferences for 119
 - settings for 116
 - highly nonlinear problem 390
 - hot colormap 470
 - hsv colormap 470
 - HTML editors 559
 - HTML formatting 466
 - HTML report 557
 - HTML tags 466
 - hue-saturation-value color model 470
 - hybrid modeling 24
 - hyperbola 48
- I**
 - ideal constraints 217
 - identical boundary meshes 360
 - identity pairs
 - boundary conditions for 253
 - illegal operations, avoiding 166
 - image export 493
 - images, size of 494
 - importing
 - COMSOL Multiphysics files 86
 - constants 140
 - DXF files 87
 - meshes 364
 - NASTRAN files 364
 - scalar expression variables 142
 - STL files 89
 - unstructured data 240
 - VRML files 89
 - inch-pound-second unit system 189
 - including default values in Model M-files 575
 - independent variables 28, 170
 - index
 - for DAEs 400
 - of a variable 401
 - inflection point 48
 - influx as integral of solution 272
 - inheritance of variables 179
 - initial conditions, using 214
 - initial values
 - evaluating for postprocessing 446
 - on boundaries 250
 - integral constraints 274
 - integral transforms 165
 - integrals, plotting as functions of time or parameter 458
 - integration
 - computing quantities using 456
 - on deformed meshes 271
 - quadrature order 457
 - types of 457
 - integration coupling variables 269
 - integration method
 - integration 457
 - summation 457
 - integration order 457
 - interactive meshing 346
 - interior boundaries 43
 - keeping 33
 - normal component direction on 172
 - removing 35
 - setting conditions on 252
 - international statute mile 198
 - interpolated data, using functions based

- on 154
 - interpolating in time 463
 - interpolation
 - of expressions for postprocessing 456
 - of external data 236
 - transfinite 320
 - using data from COMSOL Script or MATLAB 160
 - using splines 155
 - inverted mesh elements 175, 374
 - islinear operator 165
 - isosceles control polygon 47
 - isosurface levels 536
 - isosurface plots 536
 - iterative solvers 428
 - preconditioners for 429
- J** Jacobian
 - and coupling variables 294
 - avoiding contribution to 164
 - correct versus incorrect 382
 - reasons for an incorrect 382
 - symmetric 431
- Jacobian matrix 380
 - determinant of 174
 - update of 398
- Java heap space 124
- jet colormap 470
- K** kilogram calorie 198
 - knot 198
 - Krylov space, dimension of 405
- L** L2 error norm 422
 - L2 norm, for error estimate 421
 - labels
 - for geometry objects 114
 - saving preferences for 119
 - Lagrange multipliers 266
 - corresponding postprocessing variables 266
 - left preconditioning 435
 - legend, displaying in plots 477
 - lengths, displaying 77
 - letter format 557
 - lighting 104
 - ambient 110
 - directional 107
 - headlight 105
 - using point lights 108
 - using scene lights 105
 - using spotlights 109
 - line integrals, in projection coupling variables 284
 - line plots 497
 - properties for 476
 - line styles, for boundary groups 208
 - line width 120
 - linear predictor 409
 - linear system solvers 426
 - selecting 428
 - linear transformations 33
 - for extrusion coupling variables 275
 - for periodic boundary conditions 259, 260
 - for projection coupling variables 285
 - linear vs. nonlinear solver, selecting 381
 - linearization point 386, 437
 - for eigenvalues 182
 - linearized model 386
 - loads
 - at points 256
 - on edges 256
 - locked coefficients 211, 249
 - in boundary settings 251
 - log
 - from solver 450
 - saving to a file 452

- logarithmic scale in plots 466
- logical operators 147
- lookup table 154, 236
- loops, in solver scripting 444

M mapped mesh 300

- creating 320
- geometry requirements for 321
- techniques for 323

mapping solution from 2D axisymmetry to 3D 284

mass balances 15

mass moment of inertia 79

massless particles 517

material library

- adding material to 233
- creating 234
- deleting 235
- using 229
- using data from MatWeb in 235

material properties, accessing manually 232

Material/Coefficients Library dialog box 229

math symbols 468

mathematical

- constants 147
- functions 147

MATLAB 7

- exporting data to 492
- using interpolation data from 160

MATLAB figure windows, plotting in 466

MATLAB functions, using in COMSOL Multiphysics 161

MATLAB movies 480

MatWeb material property data 235

max/min markers 496

MAXHEAP variable 124

maximum element size 309

maximum values, displaying 496

mean operator 165

measurements, interpolating from 236

memory information for the Java heap 31, 124

memory usage, monitoring 452

memory-saving techniques for rendering 123

MEMS Module

- application areas 17

- application modes 17

mesh

- adaptive generation of 419

- canceling generation of 319

- converting 345

- copying 360

- creating boundary layers 338, 359

- creating in 1D and 2D 307

- creating in 3D 301

- creating interactively 346

- creating swept 333

- creating unstructured 301

- deleting 349

- elements 300

- exporting 364

- extruding 328

- identical 360

- importing 364

- optimizing quality of 304

- refining 344

- report contents for 564

- revolving 328

- statistics for 343

- visualizing 370

mesh cases 373

mesh conversion 345

mesh curvature cut off 303, 314

mesh curvature factor 303, 311

- mesh density, controlling 308
- mesh edges 300
- mesh element number, variable for 174
- mesh element scale factor 174
- mesh element size, variable for 174
- mesh faces 300
- mesh parameters
 - controlling 301
 - using on a 2D geometry 308
- mesh quality measure 174
- mesh quality, visualizing 371
- mesh refinement 344
 - adaptive 419
- mesh rendering 117, 121
- mesh sizes
 - predefined 334
- mesh sizes, predefined 302, 320, 334, 339
- mesh time derivative 179
- Mesh toolbar 346
- mesh type index, variable for 174
- mesh velocity 171
- mesh vertices 300
- mesh visualization parameters 370
- meshes
 - advancing front method 317
 - Delaunay method 317
 - quadrilateral 318
- meshing
 - algorithms 300
 - progress indication during 318
 - thin structures 342
- M-file path 161
- M-files 568, 574
- mile 198
- minimum value, displaying 496
- mirroring geometry objects 33
- mode analysis 18, 261
- model description 563, 572
 - adding to model 572
- model files, format for 568
- model images, saving 573
- Model Library models 576
- Model M-files
 - format for 568
 - resetting the command sequence in 575
 - using 574
- Model MPH-files 570
- model name 571
- model properties 563, 571
- Model Properties dialog box 563
- model settings 185
- Model Tree 223
 - Detail view 225
 - Inspect view 227
 - Overview 224
 - settings for 228
- models
 - formats for model files 570
 - plots when opening 472
 - updating 445
 - visualizing 98
- moment of inertia 77
 - tensor form and principal form 79
- momentum balances 15
- movie formats
 - AVI 479
 - codec 480
 - encoding format 480
 - GIF 479
 - QuickTime 479
- moving
 - geometry as bounding box 101
 - geometry objects 33
 - rows in tables 143
- MPa units 188

- MPH-files 568, 570
 - multidisciplinary modeling 7
 - multiphysics 6
 - multiple geometries 564
- N**
- naming conventions for variables 169
 - NASTRAN bulk data files
 - creating domain information from 366
 - importing 364
 - NASTRAN bulk data format 364
 - nautical mile 198
 - Navier-Stokes equations 15
 - Neumann boundary conditions 247
 - nojac operator 164
 - non-ideal constraints 217
 - nonlinear materials 236
 - nonlinear properties 144
 - nonlinear stationary solver 380
 - normal components, direction of on interior boundaries 172
 - normal vector
 - to element surfaces 173
 - variables for 172
 - normalization of eigensolutions 403
 - numerical integration 457
 - numerical values, displaying 459
- O**
- Object Properties dialog box 55, 76
 - opening COMSOL Multiphysics models 568
 - operators
 - adj 168
 - at 167
 - binary 147
 - depends 165
 - diff 164
 - down 165
 - fsens 168
 - if 166
 - islinear 165
 - logical 147
 - mean 165
 - nojac 164
 - pdiff 164
 - ppr and pprint 167
 - quad 166
 - reacf 168, 266
 - realdot 169
 - sens 168
 - test 164
 - unary 147
 - up 165
 - with 166
 - optimizing mesh quality 304
 - orbit 101
 - orthographic projection 103, 552
 - out-of-core PARDISO solver 427
 - output formats 556
 - direct printing 557
 - HTML 557
 - output times 395
 - from time-dependent solver 392
 - overlapping objects 128
 - overlying multiple plots 466
- P**
- panning 102, 551
 - paper formats 557
 - parabola 48
 - parabolic PDEs 434
 - parabolic problems 434
 - parabolic velocity profile 171
 - parameter values
 - accessing solution at previous 409
 - for parametric solver 407
 - parameterization variables
 - for surface boundaries 171
 - parametric segregated solver 416
 - parametric solver 405
 - distributed parameter sweeps 406

- list of parameter values for 407
- settings for 409
- stop conditions for 410
- parametric sweeps 383
 - geometric 384
- PARDISO direct solver 427
- partial differential equations, application
 - modes for 8
- particle tracing 517
- path, for M-files 161
- path-following method 406
- pathlines 517
- PDE modes 8
- PDEs
 - elliptic 434
 - parabolic 434
 - stationary 380
 - time-dependent 380
- perimeters 77
- periodic boundary conditions 257
 - implementation of 260
 - linear transformations for 259
- perspective projection 104, 552
- phase property 183
- phasor variable 183
- physical memory, current and peak usage
 - 31, 452
- physics modes 8
- physics section 560
- physics settings
 - copying 144
 - opening dialog boxes for 205
 - specifying 205
- piecewise analytic functions 151
- Plot toolbar
 - for 2D plots 501
 - for 3D models 532
- plot types 454
 - selecting 463
- plots
 - copying 551
 - editing 551, 552
 - exporting 551
 - exporting data to file 492
 - on boundaries 507
 - printing 551
 - when opening models 472
- plotting
 - in COMSOL figure windows 465
 - in MATLAB figure windows 466
 - in separate figure windows 465
 - integrals as functions of time or parameter 458
 - while solving 424, 481
- point constraints 256
- point lights 108
- point loads 256
- point settings 564
- points
 - adding to geometry 38
 - coordinates for 77
 - distance between 77
 - editing position of 55
- polynomial functions 152
- polynomial-preserving recovery 167, 469
- polynomials, using to describe data 240
- positive definite matrices 433
- postprocessing
 - axisymmetric models in 3D 496
 - expressions and quantities 454
 - for deformed meshes 455
 - selecting units for 191
 - using interpolation 456
- postprocessing plots 565
- postprocessing section, in reports 560
- ppr operator 167

- pprint operator 167
- preconditioned linear systems 435
- preconditioners 428
 - selecting 429
 - selection guidelines for 430
 - solving without 430
- predictor
 - constant 409
 - for parametric solver 409
 - linear 409
- predictor, for the initial guess 399
- preference file
 - highlighting settings 119
 - label settings 119
 - rendering settings 119
- pressure, additional units for 198
- preview
 - of models 568
 - of work planes 62
- principal axes 79
- principal plot 511
- principal stress/strain plots 510
- printing plots 551
- prism elements 301
- prism mesh, creating 328
- probe plots 481
- product updates 21
- Progress window 558
 - during mesh generation 318
 - during solution 446
 - when opening models 569
- projection 552
 - of geometry in work plane 63
- projection coupling variables 284
 - examples using 290
 - general transformations for 288
 - linear transformations for 285
- property groups 234
- publishing a report 559
- pulley, creating 3D geometry of 67

Q

- quad elements
 - See quadrilaterals, mapped mesh using and quadrilaterals, free mesh using
- quadrature 166
- quadrilateral elements 300
- quadrilaterals
 - free mesh using 318
 - mapped mesh using 320
- qual variable 174
- QuickTime movie format 479

R

- range of data to plot 468
- Rankine 195
- rational Bézier curves 48
- reac operator 168, 266
- reaction force operator 168, 266
- reaction forces
 - computing 266
 - including in solution output 442
 - integrals of 168
- realdot operator 169
- Recover check box 469
- recovery of stresses, fluxes, gradients, derivatives 469
- refining meshes 344
- regular font, indicating default values 563
- relative tolerance 392
 - in nonlinear solver 390
- reldetjac variable 174
- reldetjacmin variable 174
- release time for particles 520
- remeshing domains 349
- rendering
 - line settings 120
 - memory-saving techniques for 123
 - saving preferences for 119
 - settings for 116

- repair of geometry objects 33
- repair tolerance 33, 89
 - for DXF import 87
- report generation 555
- report sections
 - geometry 560
 - introduction 560
 - model properties 560
 - physics 560
 - postprocessing 560
 - solution 560
- resetting the model 575
- residual 385
- resolution
 - adjusting for dashed lines 477
 - for line plots 527, 546
 - of images 494
 - of narrow regions 304
- restarting the solvers 445
- results description, adding to model 572
- revolved mesh 301
- revolving
 - 2D geometries into 3D 65
 - meshes 328
- RF Module
 - application areas 17
 - application modes 18
 - material properties supported in 18
- right preconditioning 435
- Robin conditions. See generalized Neumann boundary conditions
- rotating geometry objects 33
- rotation axis 74
- rough plots 472
- rubber-band box, zooming using 100
- ruling application mode 379
- Runge-Kutta methods for solving particle equations of motion 518

S

- saving
 - COMSOL Multiphysics models 568
 - models on server 570
- scalar variables, saving and loading 142
- scaling
 - geometry before meshing 307
 - geometry objects 33
 - of images 494
 - of load and constraint symbols 119
- scene lights 105, 552
- scripting using solver commands 443
- segregated iterations 413
- segregated solver
 - parametric 416
 - stationary 411
 - time-dependent 417
- segregated substeps 413
- Select HTML Output File dialog box 557
- selecting
 - 2D geometry objects 127
 - 3D objects 127
 - a single 2D geometry object 127
 - a single 3D object 128
 - a single face element 128
 - all objects 126
 - by group 127, 305
 - current suppression 122
 - domains 126
 - methods for 126
 - multiple objects 129
 - overlapping objects 128
 - remaining domains for meshing 305
 - times steps to plot 461
- selection
 - by cycling adjacent objects 134
 - clearing of 132
 - confirmation of 130
 - conflicting 249

- indication of 127
 - of all adjacent objects 134
 - using adjacency selection methods 134
 - zooming to 100
- sens operator 168
- sensitivity analysis
 - solver settings 423
- sensitivity methods, forward and adjoint 423
- server, saving models on 570
- set difference 32
- set formula, for Boolean operations 32
- set intersection 32
- set union 32
- shape function variables 176
 - for time derivatives 178
- shift 402, 434
- shrinking coefficients in Model M-files 575
- SI units 196
- sign function 150
- simplify expressions 186
- Simpson quadrature 166
- singular mesh revolutions 332
- singularities
 - avoiding 94
 - in the geometry 94
- slice plots 534
- Smooth Parameters dialog box 469
- snap
 - to grid 31
 - to vertex 31
- solid 3D objects 67
- solid mechanics 18
- solid modeling
 - in 2D 42
 - in 3D 67
- solid objects 43
 - trimming 44
- solidifying geometry objects 31
- solution
 - interpolated 167
 - selecting for postprocessing 463
 - selecting variables in 441
 - storing and selecting for further analysis 446
- solution form
 - automatic 382
 - with coupling variables 294
- solution process
 - canceling 447
 - plotting quantities during 481
 - stop buttons for 448
- solution section, in reports 560
- solution values, displaying 459
- solutions
 - accessing for postprocessing 166
 - from a previous analysis 439
- solver algorithms 378
- Solver Manager 436
- solver scripting 443
 - commands for 443
 - solving using 444
 - using loops in 444
- solver selection, automatic 379
- solver settings 563, 564
- solvers
 - canceling 447
 - for linear systems 426
 - hierarchical structure of 447
 - in COMSOL Multiphysics 378
 - logs from 450
 - restarting 445
 - selecting 378
 - starting 445
- source transformation 278

- space coordinate variables 170
 - changing the name of 29
- space-varying properties 144
- splines, for interpolation 155
- splitting geometry objects 35
- spotlights 109
- starting the solvers 445
- static problem 385
- stationary PDE problems 380, 385
- stationary segregated solver 411
- stationary solver 385
- stationary value 380
- statistics, for mesh 343
- status bar 30
- stiffness matrix
 - correct versus incorrect 382
 - symmetric 431
- STL files
 - format of 89
 - importing 89
- Stokes equations 381
- stop conditions 398
 - for the parametric solver 410
 - for the time-dependent solver 398
- stored solution 436
- streamline plots
 - in 2D 512
 - in 3D 543
- Structural Mechanics Module
 - analysis types in 19
 - application areas 18
 - application modes 19
 - special element types in 18
- subdomain integration 456
- subdomain plots 538
- subdomain settings 564
- subdomains 35
- subsections of a report 562
- suffixes
 - for report files 558
 - in variable names 169
- summation 457
- superconvergent patch recovery 469
- suppressing display of objects 122
- surface areas 77
- surface integral 457
- surface plots 502
 - properties for 476
- sweep direction, for swept mesh 335
- swept meshes 333
 - combining with free meshes 356
 - geometry requirements for 333
 - mesh elements in 337
- symmetric Jacobian matrix 431
- symmetric matrices, when solving 432
- symmetric stiffness matrix 431
- symmetry detection 433

T

- table rows, moving and clearing 143
- tangent vectors, variables for 172
- tangential derivative variables 179, 342
- tangents 52
- technical support
 - email address for 21
 - online resources 21
- temperature units 195
- test operator 164
- tetrahedral elements 300
- text formatting and symbols 466
- text symbols 468
- thin structures, meshing 342
- time convolution integral 167
- time derivatives
 - including in solution output 442
 - variables for 178
- time integral transforms 167
- time steps

- taken by solver 395
 - time variable 144, 181
 - time, additional units for 198
 - time-dependent PDEs 380
 - time-dependent segregated solver 417
 - time-dependent solver 391
 - order of the time-stepping method in 398
 - settings for 394
 - stop conditions for 398
 - using 392
 - using the segregated 418
 - times for output 392, 395
 - times, selecting for plotting 461, 479
 - time-varying properties 144
 - tolerance
 - absolute 392
 - for time-stepping solver 392
 - relative 392
 - toolbar, in figure windows 551
 - transfinite interpolation 320
 - transformation matrix 260, 287
 - translation vector 260, 287
 - transparency 552
 - in plots 112
 - triangular elements 300
 - trimming solids 44
 - typographical conventions 4
- U**
- unary operators 147
 - undoing mesh operations 351
 - unexpected units 191, 194
 - Unicode-based characters 468
 - unit systems
 - British engineering units 189
 - CGS 188
 - EMU 188
 - ESU 188
 - FPS 189
 - gravitational IPS 189
 - IPS 189
 - MPa 188
 - SI 188
 - switching 194
 - units
 - during postprocessing 191
 - highlighting unexpected 191, 194
 - multiplying with 192
 - SI system 196
 - using 188
 - unsteady problems 391
 - unstructured data, importing 240
 - up and down directions, variables defined in 186
 - up operator 165
 - updates, checking for 21
 - updating models 445
 - URL for model information 571
 - user forums 21
 - user models, creating library for 577
- V**
- Vanka preconditioner/smoothen 430
 - variable inheritance 181
 - variable name suffixes 169
 - variables
 - available by inheritance 179
 - boundary coupled shape 180
 - for gradient components 178
 - for mesh quality 371
 - for solution output 441
 - for space coordinates 170
 - for the eigenvalue 181
 - for time 181
 - geometric 170
 - naming convention for 169
 - to solve for 439
 - vector elements, preconditioners for 431
 - vector graphics 493

- vector-valued expressions, entering 145
- velocity profile, parabolic 171
- vertex 35
- vertices
 - adding to geometry 38
 - editing position of 55
 - snapping to 31
- view
 - changing in 3D 101
 - explicit settings for 104
 - multiple geometries 36
- virtual memory, current and peak usage
 - 31, 452
- Visualization/Selection toolbar 99, 116
- visualizing
 - meshes 370
 - model geometries 98
- volume integral 457
- volumes, displaying 77
- VRML files
 - format 89
 - importing 89
- W** wave colormap 470
- wave propagation 18
- web sites related to COMSOL 21
- wedge elements. See prism elements
- weights
 - for prioritizing eigenmodes 422
 - in control polygon 48
- wireframe plot 123
- with operator 166
- work planes
 - at a specified edge angle 61
 - controlling 59
 - creating 59
 - example of using 67
 - parallel to a face 60
 - preview settings for 62
- projecting geometry in 63
- quick positioning of 60
- specify by vertices 61
- Z** zooming 100, 552
 - using the mouse or arrow keys 102

