COMSOL MULTIPHYSICS®

QUICK START AND QUICK REFERENCE

VERSION 3.4

COMSOL

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@femlab.nl
www.femlab.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
FEMLAB GmbH
Berliner Str. 4
D-37073 Göttingen
Phone: +49-551-99721-0
Fax: +49-551-99721-29
info@femlab.de
www.femlab.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
Tegnérgatan 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 284747
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 Quick Start and Quick Reference

Version: October 2007 COMSOL 3.4

CONTENTS

Chapter 1: Introduction

Chapter 2: A Quick Tour of COMSOL Multiphysics

1

Introduction

Welcome to COMSOL Multiphysics®! The purpose of this *Quick Start* is to help you get started learning how to take advantage of all the powerful features and capabilities in this state-of-the-art software package. It walks you through an example, step-by-step, and thereby introduces you to all the basic concepts that arise when creating and solving a model in COMSOL Multiphysics. If you would like additional details at any point, you can refer to the full documentation set as described on the next page. In the event the printed version is not handy, you can always access an electronic version of the full documentation set by going to the **Help** menu in COMSOL Multiphysics. We have copied several sections from the *COMSOL Multiphysics User's Guide* and some from the *COMSOL Multiphysics Modeling Guide* to avoid cross-references.

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/CD package.
- *COMSOL New Release Highlights*—information about new features and models in the 3.4 release. Included in the DVD/CD package.
- COMSOL License Agreement—the license agreement. Included in the DVD/CD 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*—the book you are reading, provides a quick overview of COMSOL's capabilities and how to access them. A reference section contains comprehensive lists of predefined variable names, mathematical functions, COMSOL operators, equation forms, and application modes.
- *COMSOL Multiphysics User's Guide*—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.
- *COMSOL Multiphysics Modeling Guide*—provides an in-depth examination of the software's application modes 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's capabilities from COMSOL Script or MATLAB.
- *COMSOL Multiphysics Reference Guide*—provided only in the form of online documentation. It reviews each command that lets you access COMSOL's functions

from COMSOL Script or 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*.

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*.

The optional COMSOL Script software comes with the *COMSOL Script User's Guide* and the *COMSOL Script Command Reference* (online only). Connected to COMSOL Script are the Optimization Lab, Signal and Systems Lab, and COMSOL Reaction Engineering Lab, which also provides an interface to the Chemical Engineering Module. For each of these products, a separate *User's Guide* provides detailed information.

Note: The full documentation is available in electronic versions—PDF and Help Desk (HTML) formats—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 product 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. 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.

As noted, the underlying mathematical structure in COMSOL Multiphysics is a system of partial differential equations. In addition to the physics mode and the modules, we provide three ways of describing PDEs through the following PDE modes:

- *Coefficient form*, suitable for linear or nearly linear models
- *General form*, suitable for nonlinear models
- *Weak form*, for models with PDEs on boundaries, edges, or points, or for models using terms with mixed space and time derivatives. (The weak form provides many additional benefits, and we review them in the context of specific models in other books in this documentation set.)

Using the application modes in COMSOL Multiphysics, you can perform various types of analysis including:

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

To solve the PDEs, 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. This book provides an introduction to multiphysics modeling in the section "Thermal Effects in Electronic Conductors" on page 36. In addition, the *COMSOL Multiphysics Modeling Guide* covers multiphysics modeling techniques in the section "Creating Multiphysics Models" on page 318. The "Multiphysics" chapter in the *COMSOL Multiphysics Model Library* also contains several examples.

Along these lines, one unique feature in COMSOL Multiphysics is something we refer to as *extended multiphysics*: the use of coupling variables to connect PDE models in different geometries. This represents a step toward system-level modeling.

Another unique feature is the ability of COMSOL Multiphysics to mix domains of different space dimensions in the same problem. This flexibility not only simplifies modeling, it also can decrease execution time.

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 3.4 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.

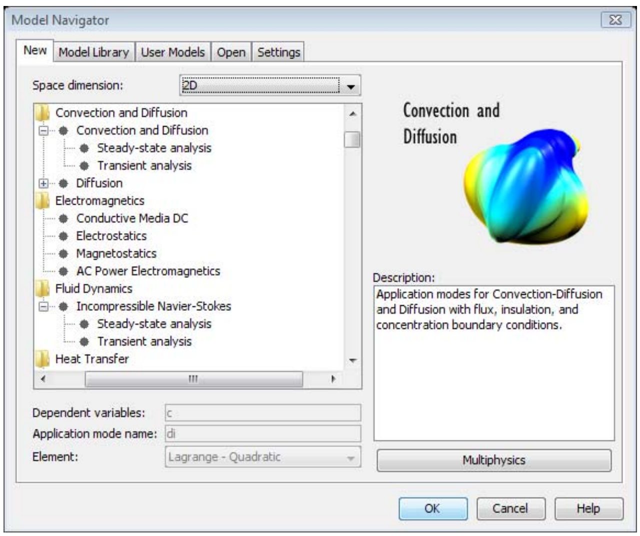
You can build models of all types in the COMSOL Multiphysics user interface. For additional flexibility, COMSOL also provides its own scripting language, COMSOL Script, where you can access the model as a Model M-file or a data structure. COMSOL Multiphysics also provides a seamless interface to MATLAB. 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 Multiphysics 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.



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, which we describe in more detail in the next section, 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
- PDEs

Axisymmetric variants are also available for most application modes. More details about the application modes appear in the following section of this book, and you can find extensive information about them in the *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 both model files installed with COMSOL Multiphysics as well as an accompanying text. Simply load one of them into the Model Navigator and hit the Solve button to watch it run. As noted earlier, each model includes extensive documentation including technical background, a discussion of the results, and step-by-step descriptions of how to set up the model.

In a short time, though, you will want to start doing your own modeling, and 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. In 2D you can import CAD data on the DXF-file format. In 3D, the software provides mesh import for meshes on the NASTRAN format and the COMSOL Multiphysics native format. In combination with the programming tools, you can even use images and magnetic resonance imaging (MRI) data to create a geometry. For other CAD formats, the CAD Import Module and its add-ons provide support for all popular types of CAD files.

When the geometry is complete and various parameters defined, COMSOL Multiphysics automatically meshes a geometry, but you take charge of the mesh-generation process by accessing 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, COMSOL Multiphysics features both direct and iterative solvers. A range of preconditioners are available for the iterative solvers. COMSOL Multiphysics sets up application mode-dependent solver defaults.

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
- Streamline plots
- Particle tracing
- Cross-section plots and numerical interpolation
- Animations
- Export of solution data to text files and the COMSOL Script workspace
- Integration on boundaries and subdomains

In some cases you might need functionality that extends beyond what we offer in COMSOL Multiphysics today. For those occasions, you can turn to COMSOL Script. Within that environment you can work with COMSOL data structures and functions using script programming. This makes COMSOL Multiphysics a powerful tool for nonstandard or multidisciplinary modeling. There is also a full interface to MATLAB.

Application Mode Overview

Solving PDEs generally means you must take the time to set up the underlying equations, material properties, and boundary conditions for a given problem. COMSOL Multiphysics, however, relieves you of much of this work. The package provides a number of *application modes* that consist of predefined templates and user interfaces already set up with equations and variables for specific areas of physics. Special properties allow the selection of, for instance, analysis type and model formulations. The application mode interfaces consist of customized dialog boxes for the physics in subdomains and on boundaries, edges, and points along with predefined PDEs. A set of application-dependent variables makes it easy to visualize and postprocess the important physical quantities using conventional terminology and notation. Adding even more flexibility, the *equation system view* allows you to easily examine and modify the underlying PDEs in the case where a predefined mode does not exactly match the application you wish to model.

Note: Suites of application modes that contain a large number of models optimized for specific disciplines are available in a group of optional products: the AC/DC Module, Acoustics Module, Chemical Engineering Module, Earth Science Module, Heat Transfer Module, MEMS Module, RF Module, and Structural Mechanics Module.

Application Modes in COMSOL Multiphysics

The following table lists COMSOL Multiphysics's application modes and their availability for 1D, 1D axisymmetric, 2D, 2D axisymmetric, and 3D geometries as well as the default application mode name (suffix for application mode variables) and dependent variable names:

APPLICATION MODE	1D	1D AXI	2D	2D AXI	3D	DEFAULT SUFFIX	DEPENDENT VARIABLES
Acoustics							
Acoustics			√	√	√	aco	p
Diffusion							
Convection and Diffusion	√	√	√	√	√	cd	c
Diffusion	√	√	√	√	√	di	c
Electromagnetics							
AC Power Electromagnetics			√	√		qa	A_z
Conductive Media DC			√	√	√	dc	V
Electrostatics			√	√	√	es	V
Magnetostatics			√	√		qa	A_z
Heat Transfer							
Convection and Conduction	√	√	√	√	√	cd	T
Conduction	√	√	√	√	√	ht	T
Fluid Dynamics							
Incompressible Navier-Stokes			√	√	√	ns	u, v, w, p
Structural Mechanics							
Plane Strain			√			ps	u,v
Plane Stress			√			pn	u,v
Axial Symmetry, Stress-Strain				√		axi	uor, w
3D Solid, Stress-Strain					√	solid3	u,v, w
Deformed Mesh							
Moving Mesh (ALE)	√	√	√	√	√	ale	x, y, z
Parameterized Geometry			√			pg	dx, dy
PDE Modes							
Coefficient form	√		√		√	c	u

APPLICATION MODE	1D	1D AXI	2D	2D AXI	3D	DEFAULT SUFFIX	DEPENDENT VARIABLES
General form	√		√		√	g	u
Weak form, subdomain	√		√		√	w	u
Weak form, boundary	√		√		√	wb	u
Weak form, edge					√	we	u
Weak form, point			√		√	wp	u
Classical PDEs							
Convection-diffusion equation	√		√		√	cd	u
Laplace's equation	√		√		√	la	u
Heat equation	√		√		√	hteq	u
Helmholtz equation	√		√		√	hz	u
Poisson's equation	√		√		√	po	u
Schrödinger equation	√		√		√	sch	u
Wave equation	√		√		√	wa	u, u_t

Note: You can change both the application mode name (suffix) and the names of the dependent variables when starting a new model. If you have more than one application mode of the same type in a model geometry, they get unique names by appending 2, 3, 4, etc. to the end of the application mode name.

As the table indicates, these application modes fall into four broad categories:

THE PHYSICS MODES

Use the physics modes to instantly access convenient templates for specific application areas. Here you can specify physical properties for models in fields such as acoustics, diffusion, or electromagnetics. Details on how to use the physics modes appear in the section "Using the Physics Modes" on page 13 in the *COMSOL Multiphysics Modeling Guide*.

THE DEFORMED MESH APPLICATION MODES

These application modes provide support for applications with moving boundaries using the Moving Mesh (ALE) application mode and for parameterized geometries in 2D. See "Deformed Meshes" on page 391 in the *COMSOL Multiphysics Modeling Guide* for more information.

THE PDE MODES

Turn to these modes to model directly with PDEs when you cannot find a suitable physics mode. With these modes you define the problem in terms of mathematical expressions and coefficients.

COMSOL Multiphysics includes three PDE modes:

- The *Coefficient form* allows you to solve linear or almost linear problems using PDEs and coefficients that often correspond directly to various physical properties.
- The *General form* provides a computational framework specialized for highly nonlinear problems. Consider using a weak form for these problems, too.
- The *Weak form* makes it possible to model a wider class of problems, for example models with mixed time and space derivatives, or models with phenomena on boundaries, edges, or points as described with PDEs. In terms of convergence rate, these modes also set a computational framework suited for all types of nonlinear problems.

For details on how to apply the PDE modes, please refer to two sections in the *COMSOL Multiphysics Modeling Guide*, specifically "PDE Modes for Equation-Based Modeling" on page 237 and "The Weak Form" on page 291.

CLASSICAL PDES

The Classical PDEs describe a suite of well-known PDEs. They are not meant to serve as templates; we include them just for fun.

Selecting an Application Mode

MODELING USING A SINGLE APPLICATION MODE

Most of the physics application modes contain stationary, eigenvalue, and dynamic (time-dependent) analysis types. As already mentioned, these modes provide a modeling interface that lets you perform modeling using material properties, boundary conditions, and initial conditions. Each of these modes comes with a template that automatically supplies the appropriate underlying PDE.

If you cannot find a physics mode that matches a given problem, try one of the PDE modes, which allow you to define a custom model in general mathematical terms. Indeed, COMSOL Multiphysics can model virtually any scientific phenomena or engineering problem that originates from the laws of science.

MODELING MULTIPHYSICS OR SYSTEMS WITH SEVERAL DEPENDENT VARIABLES

When modeling a real-world system, you often need to include the interaction between different kinds of physics. For instance, the properties of an electronic component such as an inductor vary with temperature.

To solve such a problem, combine two or several application modes into a single model by using the multiphysics features. For the example just mentioned, combine the Conductive Media DC and Heat Transfer by Conduction modes. In this way you create a system of two PDEs with two dependent variables: V for the electric potential and T for temperature.

Combining physics modes and PDE modes also works well. Assume, for instance, that you want to model the fluid-structure interactions due to the vibrations of yoghurt in a cardboard container as it rides on a conveyor belt. You could start with the Plane-Stress mode in the Structural Mechanics Module to model the container walls, then add a PDE to model the irrotational flow of the fluid. This approach also creates a system of two PDEs but requires that you define one of them from scratch, in this case using the general PDE formulation.

To summarize the proposed strategy for modeling processes that involve several types of physics: Look for application modes suitable for the phenomena of interest. If you find them among the physics modes, use them: if not, add one or more PDE modes.

The approach to couple multiple application modes is to use dependent variables or their derivatives, or use expressions containing the dependent variables. The coupling can occur in subdomains and on boundaries. In the example above, use the dependent variable T for the temperature of the conductivity σ in the Conductive Media DC application mode:

$$\sigma = \frac{1}{[\rho_0(1+\alpha(T-T_0))]}$$

For this case, which is an example of Joule heating, and some other common multiphysics couplings, there are predefined combinations of application modes with the typical couplings available as the default settings (see the **Predefined Multiphysics**

Couplings folder in the Model Navigator). For a complete description of this models, see the section “Example: Resistive Heating” on page 334 in the *COMSOL Multiphysics Modeling Guide*.

Internet Resources

For more information about the COMSOL products, including licensing and technical information, a number of Internet resources are available. 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 many technical questions and license questions: http://www.comsol.com/support/knowledgebase/

Product updates: http://www.comsol.com/support/updates/

CONTACTING COMSOL BY EMAIL

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

Send 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 the COMSOL user community shares solutions and tips on COMSOL modeling.

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 COMSOL office, or send your questions to support@comsol.com.

2

A Quick Tour of COMSOL Multiphysics

This chapter provides a quick tour of COMSOL Multiphysics through an overview of the basic procedures and an example model. The discussion covers the entire modeling process using a step-by-step approach that includes many important aspects of the software:

- The use of predefined physics application modes
- The definition of a multiphysics problem
- The possibility to model using physical properties that depend on the solution itself
- The extraction of design parameters using postprocessing tools

In addition to these COMSOL Multiphysics features, this first example covers general modeling methods:

- Reduction of the dimension of parts of the problem from three to two space dimensions
- Checking simulation results to validate the solution

When creating a model in COMSOL Multiphysics, the typical modeling steps include:

1 Creating or importing the geometry
2 Meshing the geometry
3 Defining the physics on the domains and at the boundaries
4 Solving the model
5 Postprocessing the solution
6 Performing parametric studies

The following section describes the main components and procedures in COMSOL Multiphysics and its operating modes.

Basic Procedures

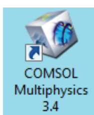
COMSOL Multiphysics supplies a number of easy-to-use tools and commands to help with modeling and analysis. After mastering the basic procedure in this chapter, you will want to learn the details about of the various functions and features.

Starting COMSOL Multiphysics

The very first step is to install COMSOL Multiphysics on your computer according to the instructions in the *COMSOL Installation Guide*.

STARTING COMSOL MULTIPHYSICS IN WINDOWS

The easiest way to start COMSOL Multiphysics is to double-click the COMSOL Multiphysics 3.4 icon.



The COMSOL Multiphysics 3.4 icon on the Windows desktop.

You can also start COMSOL Multiphysics from the Windows Start menu. See the *Installation Guide* for information about starting COMSOL Multiphysics with a server connection and using COMSOL Script or MATLAB.

STARTING COMSOL MULTIPHYSICS IN LINUX AND UNIX

To start COMSOL Multiphysics, run the command

```
comsol
```

at the system command prompt.

STARTING COMSOL MULTIPHYSICS IN MAC OS X

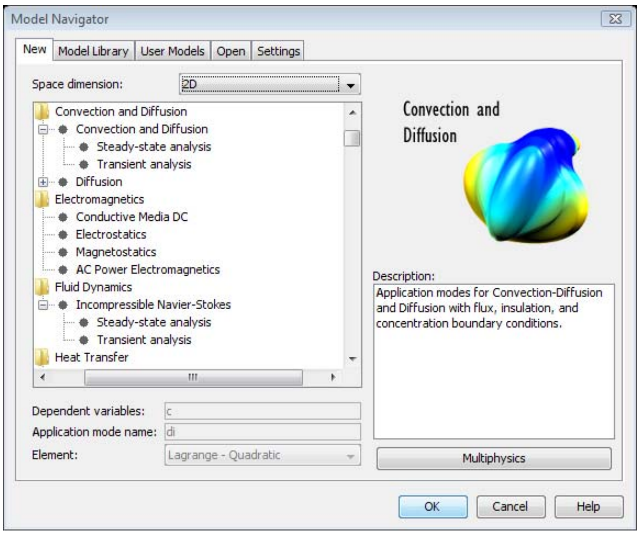
Double-click the COMSOL Multiphysics icon in the COMSOL34 folder to launch COMSOL Multiphysics.

The COMSOL Multiphysics icon in Mac OS X.

Creating and Opening Models

When COMSOL Multiphysics starts, the first window that appears is the **Model Navigator**. This is a starting point where you create new models and open your own models or those from the model libraries. From the **Model Navigator** you enter the main COMSOL Multiphysics application window to start building a model or continue working with an existing model.

STARTING A NEW COMSOL MULTIPHYSICS MODEL



1. In the **Model Navigator**, select the space dimension from the **Space dimension** list. Choose from 1D, 2D, and 3D geometries using Cartesian coordinates and, for most application modes, 1D and 2D axisymmetric geometries using cylindrical coordinates.
2. Select an application mode from the list of application modes to the left.

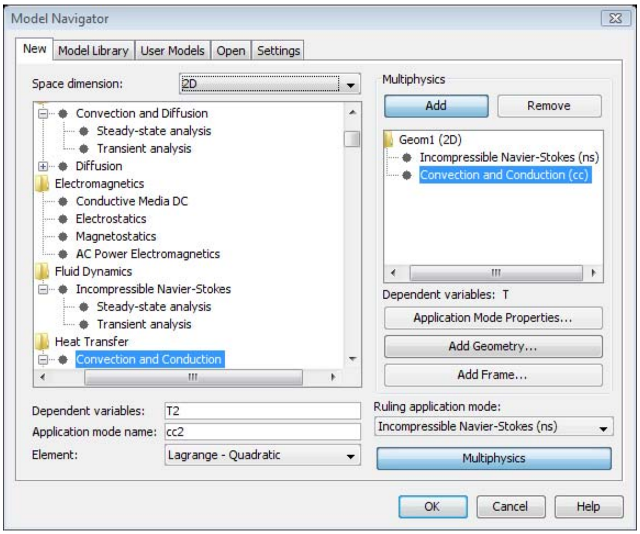
   Opening a folder at the top level shows all application modes within that application area. For some application modes you can select from different analysis types or solver types by clicking the plus sign at the application mode node.
3. Click **OK**.

OPENING AN EXISTING COMSOL MULTIPHYSICS MODEL

To open a COMSOL Multiphysics model anywhere on the file system, click the **Open Model** tab, select your Model MPH-file, and click **OK**. You can also open models from the main COMSOL Multiphysics window by choosing **Open** from the **File** menu. When browsing for models, a preview of the model appears if you activate the **Preview** button. To save a preview image for your model, choose **Save Model Image** from the **File** menu before saving the model.

To open a Model MPH-file (`.mph`) directly in Windows, double-click its icon from, for example, Windows Explorer. That opens COMSOL Multiphysics and loads the model.

CREATING A MULTIPHYSICS MODEL



Use the following steps to create a multiphysics model with several types of physics and equations:

1. Click the **Multiphysics** button.
2. Make an appropriate selection from the list of application modes.
3. Click **Add** to add it to the current model.
4. Continue adding application modes by selecting them and then clicking the **Add** button.
5. For extended multiphysics modeling using more than one geometry, click the **Add Geometry** button to specify a new geometry. Then proceed to add application modes to the new geometry.
6. Click **OK**.

It is also possible to open the **Model Navigator** from the **Multiphysics** menu to add and remove application modes in the model. See the "Multiphysics Modeling" chapter in the *Modeling Guide* for more information about multiphysics modeling.

Note: If you use the entries in the folders for predefined multiphysics couplings in the Model Navigator (for example, **Electro-Thermal Interaction**, COMSOL Multiphysics starts a model with the appropriate application modes, including predefined couplings as the default settings or as parts of available predefined boundary conditions and subdomain settings.

STARTING A MODEL WITH A GEOMETRY ONLY

It is also possible to start COMSOL Multiphysics without any physics or equations. That way you can create a model geometry and later add physics, equations, and additional geometries by opening the Model Navigator from the Multiphysics menu. To open a model with a geometry only, use these steps:

1. Click the **New** tab in the **Model Navigator**.
2. Select the space dimension of the geometry from the **Space dimension** list.
3. Click **OK** without making a selection in the list of application modes.

If you already have selected an application mode, you can clear the selection by **Ctrl**-clicking on the selected application mode in the list.

DISABLING AND ENABLING THE MODEL NAVIGATOR AT STARTUP

You can disable the Model Navigator so that the main window opens directly at startup. To do this from the Model Navigator, click the **Settings** tab and clear the **Open Model Navigator on startup** check box. You can change this setting from the main application window by opening the **Preferences** dialog from the **Options** menu and selecting the **Open Model Navigator on startup** check box again on the **General** tab.

Using Commands and Dialog Boxes

In the COMSOL Multiphysics graphical user interface you use commands, toolbars, and dialog boxes for every task.

UNDO AND REDO COMMANDS

To undo and redo many commands, choose **Undo** and **Redo** from the **Edit** menu or press Ctrl+Z or Ctrl+Y. In many cases, you can undo and redo several modeling steps.

TO CHOOSE A COMMAND FROM A MENU

Choose commands from the menus in the main menu bar. For many commands shortcut keys appear on the menu to the right of the command. Learning to use these shortcut keys can help you work faster. You can, for example, press Ctrl+S to save a model and press F8 to open the **Subdomain Settings** dialog box.

TO USE A DIALOG BOX

For most inputs, COMSOL Multiphysics provides dialog boxes that stay open while you work on your model. They typically contain three buttons for using or discarding your input:

- Click **Apply** to apply all inputs to the current model without closing the dialog box.
- Click **OK** or press **Enter** to apply all inputs to the current model and close the dialog box.
- Click **Cancel** or press **Esc** to discard all inputs and close the dialog box.

Extended Edit Fields

In the dialog boxes where you specify the physics, equations, and variables, an extended edit field appears if the entry that you type does not fit inside the regular edit field. It is possible to edit both the extended edit field and the regular edit field. If you do not want to use the extended edit fields, you can turn them off:

1. From the **Options** menu, open the **Preferences** dialog box.
2. On the **General** tab, clear the **Show extended edit fields** check box.
3. Click **OK**.

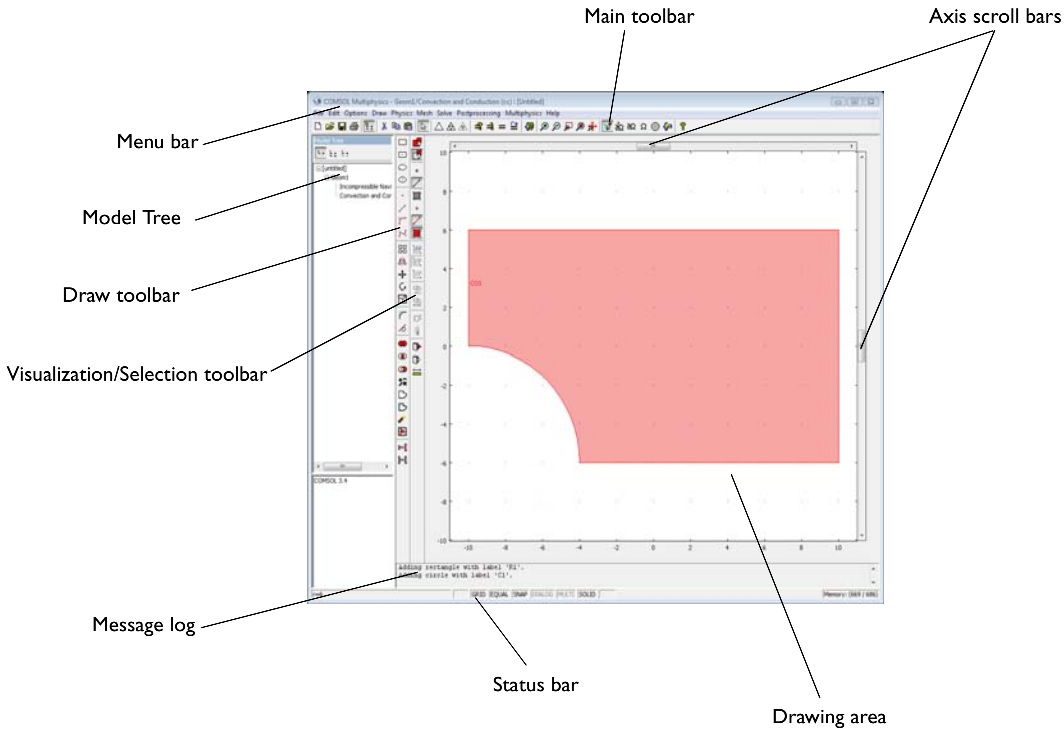
Overview of the User Interface

The COMSOL Multiphysics application window provides a graphical user interface (GUI) that handles all aspects of the modeling process:

- Preprocessing and CAD
- Specification of the physics through equations, material data, boundary conditions, couplings, and other properties
- Meshing, assembly, and solution of the finite element model
- Postprocessing and visualization of the solution and other quantities

The 1D and 2D Graphical User Interface

The figure below shows the 2D graphical user interface with annotations that highlight some of its most important elements.

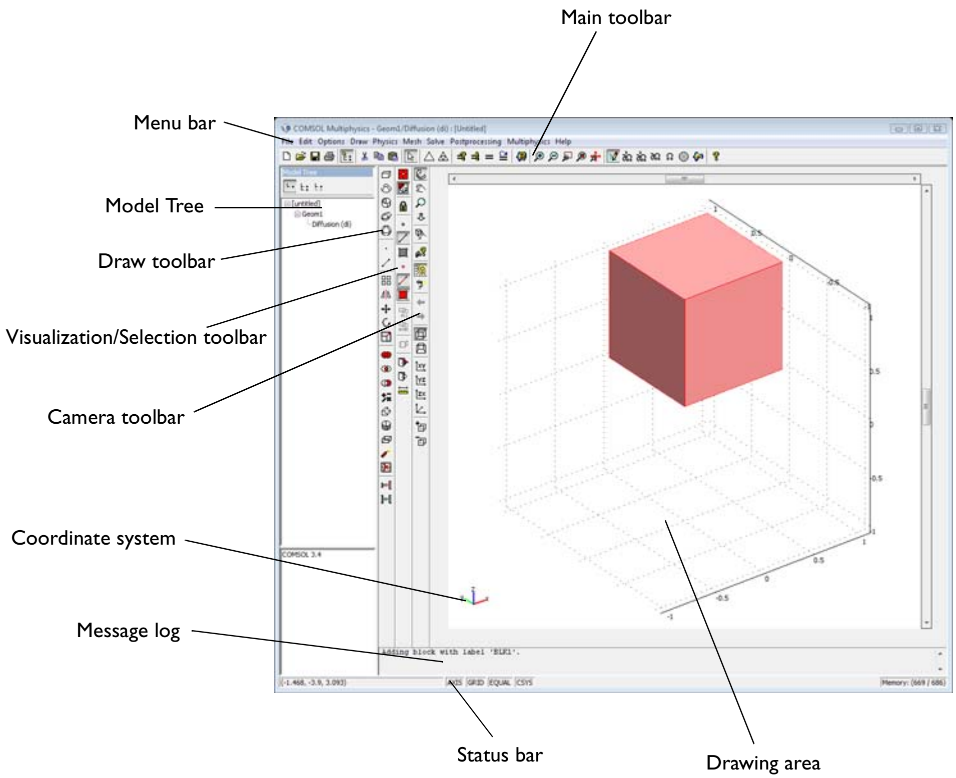


The GUI looks similar when working with 1D geometries but with fewer options in the Draw toolbar.

The 3D Graphical User Interface

The main differences between the 3D and the 1D and 2D user interfaces appear in geometry modeling, visualization possibilities of a 3D geometry, and methods for selecting objects.

The following plot shows the 3D graphical user interface together with annotations pointing out its most important parts:



The Main GUI Components

The following are the most important GUI components used during the modeling process:

- The main menu contains commands for all modeling tools.
- The Main toolbar provides quick access to the most important modeling features.
- The Draw toolbar contains CAD tools for drawing and editing geometry objects and tools for creating pairs and imprints to connect the parts of assemblies.
- The Plot toolbar provides quick access to the most common visualization methods.

- The Visualization/Selection toolbar is available for preprocessing of models. It provides tools for rendering and selection of domains.
- The Mesh toolbar contains interactive tools for creating structured and unstructured meshes.
- The Camera toolbar is available for 3D models and during the postprocessing of 2D models. It contains tools for changing the view and adding light sources.
- The drawing area displays the model during various modeling stages. You can point-and-click to select parts of the model for specification of input data. You can also click-and-drag to draw and edit 1D and 2D geometry objects.
- The color scales map colors in the plots to the numerical values of the plotted properties.
- The coordinate system is available for 3D models and during the postprocessing of 2D models. The small coordinate system adapts to the current 3D view and helps you keep track of the axis directions.
- The Model Tree provides an overview and more detailed views of the current model. It consists of two areas. In the top area you find a hierarchical list of model properties. Select an item in this list to display information about that part of the model in the bottom area of the Model Browser. By right-clicking any of the entries in this list, you can open the corresponding dialog boxes and access other commands for the settings in any application mode, geometry, or model. By accessing dialog boxes from the Model Tree, you do not need to change the current geometry or application mode. Right-clicking the top node, which represents the entire model, provides options to open the **Model Navigator**, the **Model Properties** dialog box, save the model, and much more. For more information about the Model Tree, see "Using the Model Tree" on page 217 in the *COMSOL Multiphysics User's Guide*.
- The message log displays messages about changed model parameters, geometry modeling commands, progress and convergence of the solvers, and much more. When postprocessing 1D and 2D models, you can point-and-click to get numerical values of the plotted property. The values appear in the message log. Simply scroll the message log to look at old messages.
- The status bar, located just below the message log, provides details about the current state of your COMSOL Multiphysics session:
  - The cursor coordinates appear on the far left.
  - A set of buttons immediately to the right of the cursor-coordinate field indicate the status for properties such as whether axes (**AXIS**), grid (**GRID**), and coordinate

system (**CSYS**) are displayed in the drawing area, or whether the axis scales are equal or not (**EQUAL**). Double-click such a status bar button to toggle its state.

- In the **Memory** field on the far right, current and peak virtual memory usage (both expressed in MB) are shown. Place the cursor over this area to display a tooltip that additionally displays the corresponding numbers for physical memory usage.
- When running COMSOL Multiphysics with a server, server name and port number are displayed to the left of the **Memory** field.

TOOLTIPS FOR THE TOOLBAR BUTTONS

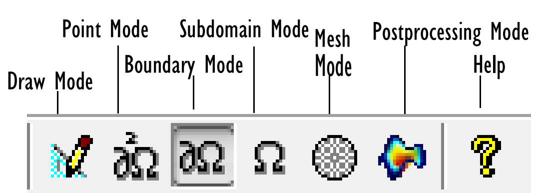
All the toolbar buttons have tooltips, which are small labels that contain a description of the command that you activate by clicking the toolbar button. The tooltips appear when you move the cursor over the toolbar button. To turn off the display of tooltips:

1. From the **Options** menu, open the **Preferences** dialog box.
2. On the **General** tab, clear the **Show tooltips** check box.
3. Click **OK**.

NAVIGATING THE SELECTION AND DISPLAY MODES

COMSOL Multiphysics contains various selection and display modes. They provide selection and display of the relevant part of the model. The following modes are available:

- Draw mode
- Point selection mode
- Edge selection mode (3D only)
- Boundary selection mode
- Subdomain selection mode
- Mesh mode
- Postprocessing mode



The mode navigation buttons in a 2D model.

Select the modes using the mode navigation buttons on the main toolbar or from a menu. To access the selection modes for points, edges, boundaries, and subdomains, point to **Selection Mode** in the **Physics** menu and then click the desired selection mode. Select Draw mode, Mesh mode, and Postprocessing mode from the corresponding menus. The mode navigation buttons provide quick mode selection and a visual indication of the current mode. In 3D, the **Point mode** button has the symbol $\partial^3\Omega$, and the **Edge mode** button uses the same symbol $\partial^2\Omega$ as the **Point mode** in 2D.

Getting Help

To open the COMSOL online documentation, use any of these methods:

- From the **Help** menu, choose **Help Desk (HTML/PDF)**.
- Press F1.
- Click the **Help** button.

In most dialog boxes you can click a **Help** button that opens the documentation that describes the functionality of the dialog box.

The COMSOL Help Desk contains the entire COMSOL documentation set. Search tools make it easy to find the desired information. The documentation set is also available as PDF files. If you are connected to the Internet you can also access our online support resources:

From the **Help** menu, choose **Online Support** or point to **Online Resources** and then click **Support Knowledge Base** to access the online COMSOL support resources. See "Internet Resources" on page 20 for more information about the COMSOL Internet resources.

Saving Models

You can save models in different formats:

- Model MPH-file, using extension `.mph`. This is the standard COMSOL Multiphysics model data format, which contains both text and binary data.
- Model M-file, using the extension `.m`. This creates an M-file, which is a text file that can run in the COMSOL Script environment or in the MATLAB environment and that you can extend using the COMSOL programming language or MATLAB commands. You can also load Model M-files into the COMSOL Multiphysics user interface.

To save a model, choose **Save** from the **File** menu or press Ctrl+S. If it is the first time that you save a model, COMSOL displays the **Save Model** dialog box. Select the file format, enter a new name for the model, and click **Save**.

Local Language Support

You can select a local language for the COMSOL Multiphysics user interface. COMSOL Multiphysics 3.4 includes the following languages:

- English
- Traditional Chinese
- Simplified Chinese
- French
- German
- Italian
- Japanese
- Korean
- Swedish

To change the language from English to another language when you start COMSOL Multiphysics, do the following steps:

1. Start COMSOL Multiphysics.
2. In the Model Navigator, click the **Settings** tab.
3. Select the language for the user interface from the **Language** list.
4. Click **OK**.

If COMSOL Multiphysics is open, you can switch the language using the **Preferences** dialog box:

1. From the **Options** menu, choose **Preferences**.
2. Select the language for the user interface from the **Language** list.
3. Click **OK**.

Note: When changing to a non-Latin language, you may have to restart COMSOL Multiphysics to get a proper display of the new language. In those cases, COMSOL Multiphysics displays a message about the need to restart.

The selected language becomes the default language until you change the language again.

DIAGNOSTIC MESSAGES

The diagnostic messages also appear in the local language. Notice that all error messages have a unique number. Refer to this number if you have questions regarding one of these messages. The Diagnostics section in the *COMSOL Multiphysics Command Reference* contains more information about the diagnostic messages.

Modeling in COMSOL Multiphysics

The model discussed later in this chapter, as well as the models in the *Modeling Guide* and the *Model Library*, all use a common format for their descriptions.

Style Conventions for the Model Descriptions

The basic flow of actions is indicated by the order of the toolbar buttons and the menus. You work from left to right when defining, solving, and postprocessing a model. Considering this, a certain style convention, or format, is used throughout the documentation for describing models. The format includes headlines corresponding to each major step in the modeling process. These headlines also roughly correspond to the different modes and menus in COMSOL Multiphysics.

MODEL NAVIGATOR

This section explains how to start a new model by selecting application modes and specifying variable names and other model properties in the **Model Navigator**.

OPTIONS AND SETTINGS

This section covers basic settings, for example, the axis or grid spacing settings. These can usually be made with commands from the **Options** menu or by double-clicking on the status bar. Use the **Constants** dialog box to enter model parameters.

GEOMETRY MODELING

Here you create the model geometry using the CAD tools on the draw menu and the draw toolbar.

PHYSICS MODELING

In this step you enter all the descriptions and settings for the physics and equations in the model. This part contains one or more of the following sections:

Subdomain Settings

In this section you specify subdomain settings. They describe material properties, sources, and PDE coefficients on the subdomains. On the subdomains it is also possible to specify initial condition and element types.

Boundary Conditions

Here you specify boundary and interface conditions.

Edge Settings

Here you specify edge settings. They describe material properties and PDE coefficients on edges (3D models only).

Point Settings

Here you specify point settings. They describe properties and values for point sources and other values that apply to geometry vertices (2D and 3D models).

Application Scalar Variables

Some application modes use scalar variables that are independent of the geometry, for example, the frequency.

Application Mode Properties

Depending on the application mode, you can change a number of its properties such as analysis type and equation formulations.

Coupling Variables

In extended multiphysics models and for certain applications, you can assign coupling variables that connect variables and expressions in various domains. In this section, you define these coupling variables.

MESH GENERATION

Here you create the finite element mesh for the model geometry. Normally you simply click the mesh buttons on the Main toolbar. In some cases you must use other commands on the **Mesh** menu to customize the mesh.

COMPUTING THE SOLUTION

Often it suffices to click the **Solve** button on the Main toolbar. Sometimes you might want to change some solver properties or analysis settings in the **Solver Parameters** dialog box or the **Solver Manager** dialog box.

POSTPROCESSING AND VISUALIZATION

Here you make the visualization settings as well as perform various postprocessing of the analysis results. You work with the **Plot Parameters** dialog box and other visualization and postprocessing tools.

Thermal Effects in Electronic Conductors

The purpose of this model is to introduce you to the general concepts of multiphysics modeling as well as the methodology of solving such problems with COMSOL Multiphysics.

The phenomena this example studies involve the coupling of thermal and electronic current balances. In this case the modeled device conducts direct current. The ohmic losses due to the device's limited conductivity generate heat, which increases the conductor's temperature and thus also changes the material's conductivity. This implies that a 2-way multiphysics coupling is in play: that is, the electric current balance influences the thermal balance and vice versa.

The modeling procedure consists of these basic steps:

- Draw the device
- Define the physics, where you specify material properties and boundary conditions
- Create a mesh
- Select and run a solver
- Postprocess the results

COMSOL Multiphysics contains an easy-to-use CAD tool, which this exercise introduces you to. You might prefer to work with some other CAD tool and then import a geometry into COMSOL Multiphysics; in that case you can skip the drawing part of the following exercise and instead start by loading an example CAD file for this problem (supplied with this software) and continue from there.

Introduction

Figure 2-1 shows the geometry of the modeled device, which is essentially half of the support frame for an IC and that is soldered to a printed-circuit board. It has two legs soldered to the pc board, and its upper part is connected to some IC through a thin conductive film.

Each conductor (leg) is made of copper, while the solder joints are made of an alloy consisting of 60% tin and 40% lead.

The model calculates the temperature assuming that the conductors must transfer 1A from the solder joints at the circuit board to the IC connected to the film.

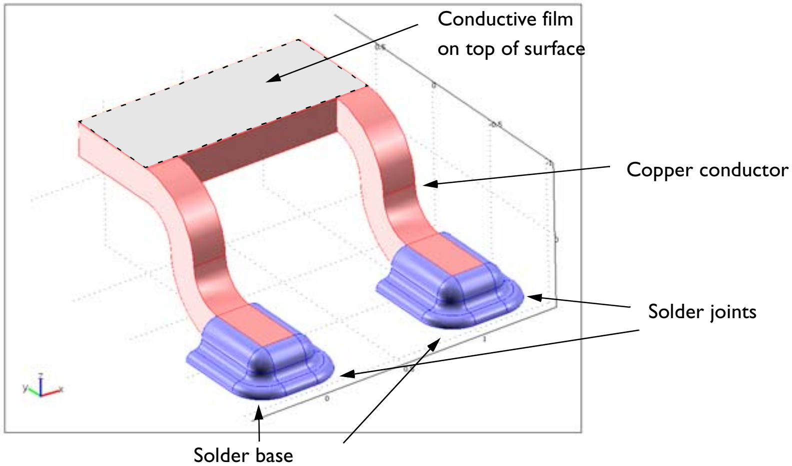


Figure 2-1: Geometry of the modeled device.

Model Definition

The electronic current balance is defined by the equation

$$\nabla \cdot (-\sigma_{\mathrm{metal}} \nabla V) = 0$$

where σ_{metal} denotes the electric conductivity of the respective alloys (in S/m) and V the electric potential (V). The conductivity is a temperature-dependent function given by the expression

$$\sigma_{\mathrm{metal}} = \frac{1}{\rho_0(1 + a(T - T_0))}$$

where ρ_0 is a reference resistivity (Ω·m) at a reference temperature T_0 (K), and a is proportionality constant (K^{-1}) for the temperature dependence.

The thermal energy balance equation includes heat production arising from the losses in the electronic conductors:

$$\nabla \cdot (-k_T \nabla T) = Q_{\mathrm{electric}}$$

where the heat source is

$$Q_{\mathrm{electric}} = \sigma_{\mathrm{metal}}|\nabla V|^2$$

In these expressions, k_{T} denotes the thermal conductivity (W/m·K) for the respective alloy, and $Q_{electric}$ is the heat source (W/m^3).

The boundary conditions for the electronic current balance are of three types:

- At the solder-joint bases, where the joints make contact to the circuit board, the potential is set to a given value

$$V_0 = \Delta V_{\mathrm{device}}$$

- The boundary condition for the thin oxide film at the top surface of the device gives the current density as a function of the potential difference over the film

$$(-\sigma\nabla V) \cdot \mathbf{n} = \kappa(V - V_{\mathrm{g}})$$

  where $\mathbf{n}$ denotes the outward pointing normal vector, κ equals the film's conductance (S/m^2) and V_{g} is the ground potential (0 V).

- All other boundaries are electrically insulating expressed with the equation

$$(-\sigma\nabla V) \cdot \mathbf{n} = 0$$

The boundary conditions for the thermal energy balance are insulating for both the film surface and the solder-joint bases so that

$$(-k_T\nabla T) \cdot \mathbf{n} = 0$$

You can assume that all other surfaces are in contact with the surrounding air and are cooled by natural convection as described with the expression

$$(-k_T\nabla T) \cdot \mathbf{n} = h(T - T_{\mathrm{amb}})$$

where h is the heat transfer coefficient (W/m^2).

Material properties for both the copper and the Sn-Pb solder alloy are in the COMSOL Multiphysics materials library.

Results and Discussion

Figure 2-2 shows the potential distribution at a voltage difference of 0.1 mV between the solder joints and the outer surface of the film. As expected, the losses appear in the narrow legs.

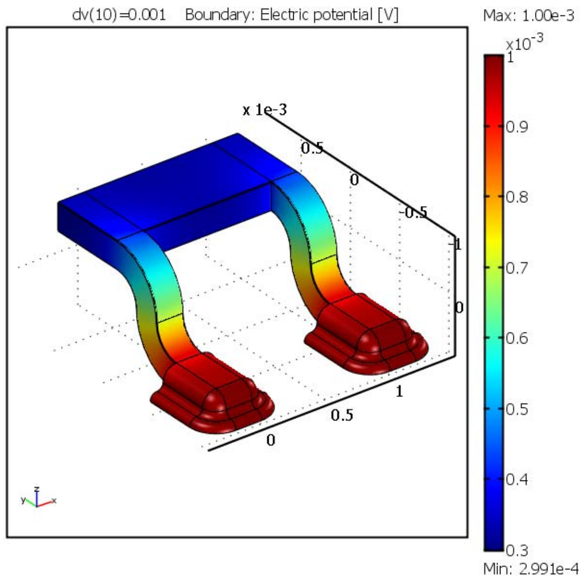


Figure 2-2: Electric potential (in V) on the surface of the device.

Figure 2-3 shows the device's temperature for a total current load of approximately 1.1 A and where the ambient temperature equals 343 K.

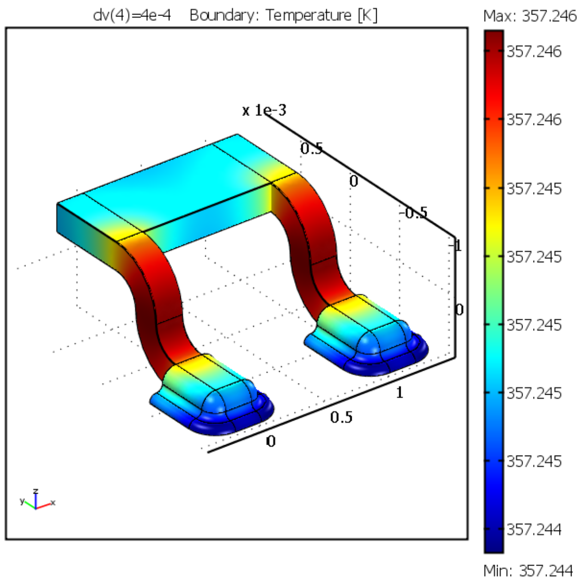


Figure 2-3: Temperature distribution in the device (in K).

The temperature field is almost uniform due to the high thermal conductivity of copper and the solder alloy. However, the device's temperature is roughly 14 K higher than the ambient.

The temperature increases exponentially with the potential difference over the device. Figure 2-5 shows the temperature as a function of the potential across the device. A total current of roughly 1.1 A flows at a total potential of 0.4 mV across the device. Figure 2-6 compares the two cases.

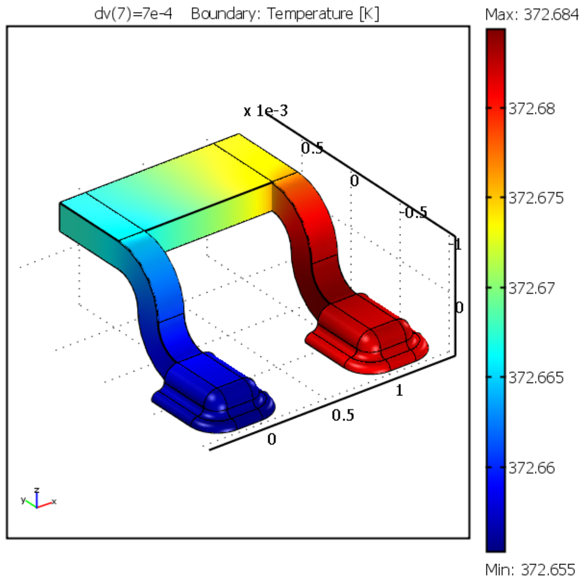


Figure 2-4: Temperature profile at about 1.1 A.

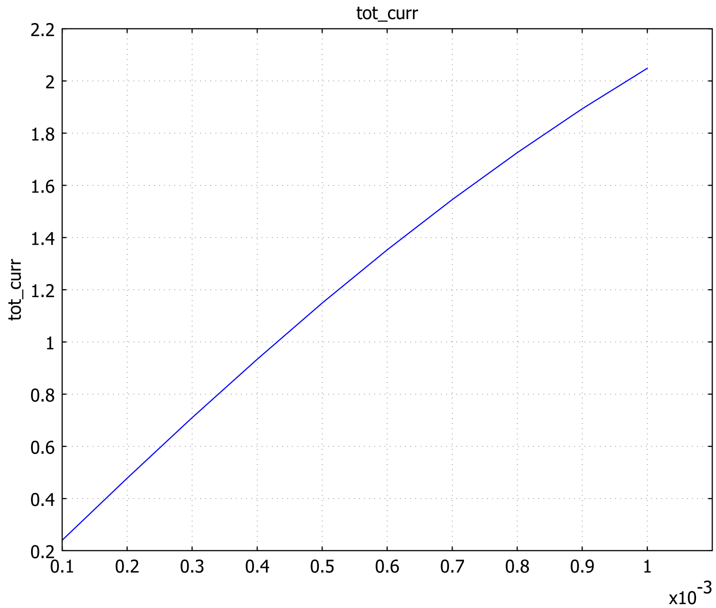


Figure 2-5: Temperature as a function of the potential difference across the device.

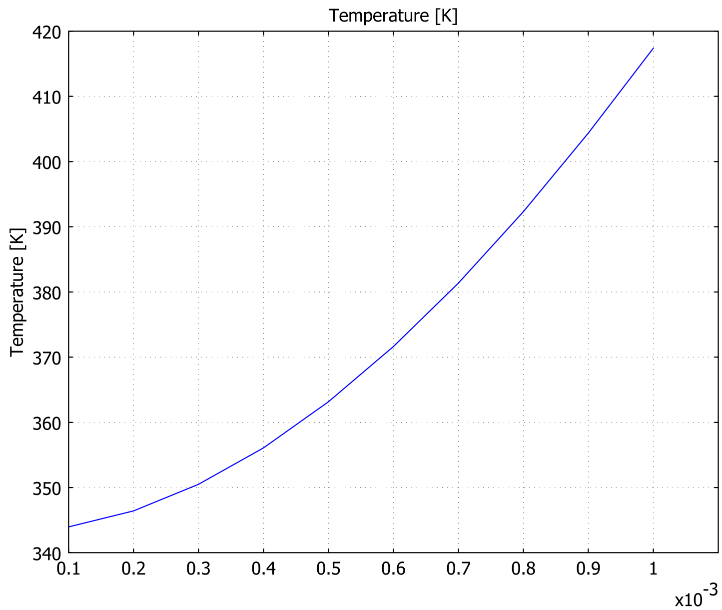


Figure 2-6: Total current as a function of the potential difference across the device.

Modeling in COMSOL Multiphysics

The implementation described in the following section uses a predefined modeling interface (in COMSOL Multiphysics called an application mode) for the multiphysics modeling of electronic current and thermal energy balances. In addition, an optional part of this exercise solves the model using a parametric solver that changes the value of the potential difference across the device using selected values.

This model is available as a ready-to-run Model MPH-file in the COMSOL Multiphysics Model Library. You have access to this model and all other models in this library from the **Model Library** tab in the Model Navigator. The following path provides the name and location of the model file:

Model Library path:

`COMSOL_Multiphysics/Multiphysics/electronic_conductor`

Modeling Using the Graphical User Interface

The modeling procedure consists of seven main steps:

1 Select an appropriate *application mode*, one of a collection of predefined modeling interfaces optimized for the modeling of a specific type of phenomena. In this case, the required application mode—which you select in the **Model Navigator**—models an electric current balance coupled to a thermal balance through heating caused by ohmic losses in an electric conductor.
2 Define constants and other input data for the model. In this second step you can also manipulate other entities related to the modeling environment, for example, the size of the drawing table and its grid. You access functionality of this type primarily in the **Options** menu.
3 Draw the model geometry using operators in the **Draw** menu.
4 Determine properties and the phenomena that take place inside the modeled device (using the **Subdomain Settings** dialog box) and how the device interacts with the surrounding environment (using the **Boundary Settings** dialog box). Both of these dialog boxes are under the **Physics** menu.
5 Create a mesh, a procedure you control with selections in the **Mesh** menu.
6 Configure settings in the **Solver Parameters** dialog box, which you access through the **Solve** menu. In this example you select a stationary solver.
7 Evaluate simulation results. A large number of evaluation methods appear in the **Postprocessing** menu.

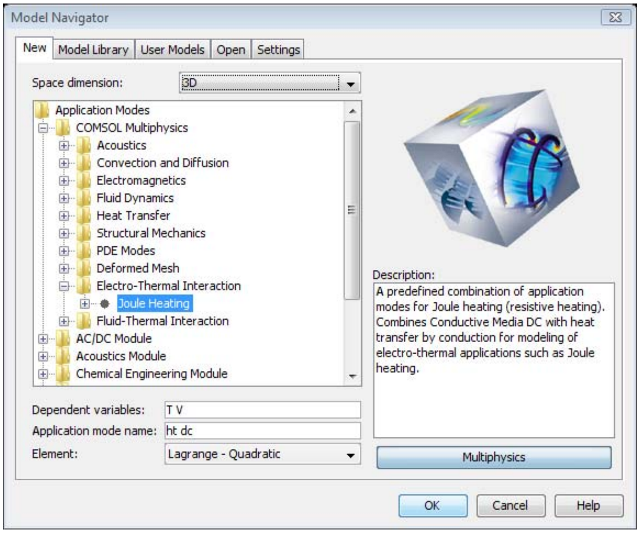
In addition to the basic steps mentioned above, you can also find the instructions for running parametric analysis.

You can now start the modeling session.

MODEL NAVIGATOR

1 Double-click on the COMSOL Multiphysics icon on the desktop to open the **Model Navigator**.
2 Go to the **New** page, and in the **Space dimension** list select **3D**.
3 Double-click the **COMSOL Multiphysics** folder to expand it.
4 Double-click the **Electro-Thermal Interaction** folder.

5 In the list of application modes, select **COMSOL Multiphysics> Electro-Thermal Interaction>Joule Heating** (see the following figure).



6 Click the **Settings** tab. In the **Unit system** list make sure that **SI** is selected.

7 Click **OK**.

OPTIONS AND SETTINGS

You are now ready to define a few constants needed as input data.

1 From the **Options** menu select **Constants**.

2 In the resulting dialog box define a reference temperature. Go to the first row. In the **Name** column type `Tr`; in the **Expression** column type `273.15[K]`, where `[K]` defines the unit, in this case kelvin for temperature; and in the **Description** column type `Reference temperature`.

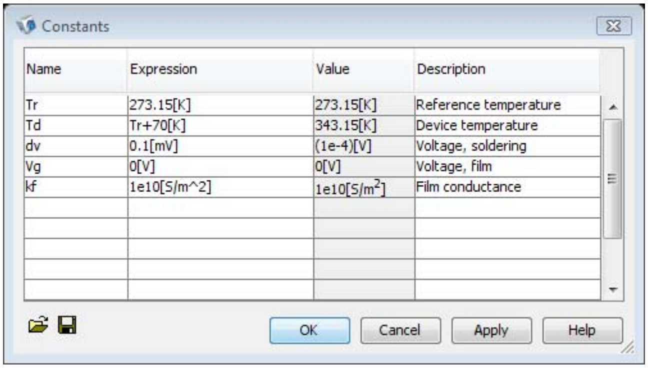
   The **Name** column defines the name that you must use when referring to this constant elsewhere in the model; the entry in the **Expression** column calculates its **Value**; and the **Description** column contains your notes regarding this constant. Any constant can be a function of other constants, which is why the **Expression** column is not always identical to the **Value** column. Also the value in the **Value** column is presented in the relevant unit of the selected base unit system. For example, if you instead enter `0[degC]` or `32[degF]` to define the reference temperature in degrees Celsius or degrees Fahrenheit, respectively, the **Value** column still displays **273.15[K]**, because kelvin is the unit for temperature in the SI unit system.

3 Go to the second row. In the **Name** column enter `Td`; in the **Expression** column enter `Tr+70[K]`; and in the **Description** column enter `Device temperature`.

4 Continue making entries in the **Constants** list:

- **Name** `dv`; **Expression** `0.1[mV]`; and **Description** `Voltage, soldering`.
- **Name** `Vg`; **Expression** `0[V]`; and **Description** `Voltage, film`.
- **Name** `kf`; **Expression** `1e10[S/m^2]`; and **Description** `Film conductance`.

5 The **Constants** dialog box should now look like the following figure:



6 Click **OK**.

GEOMETRY MODELING

Now create the model geometry starting with 2D work planes in which you draw projections, then extrude and revolve these projections to create a 3D object. If you want to load the geometry and proceed directly to the physics settings, skip this section and continue with the section “Physics Settings” on page 60.

Defining a Work Plane using the Quick Page

1 From the **Draw** menu select **Work-Plane Settings**.

2 Go to the **Quick** page, then click the **y-z** option button.

3 Click the **Apply** button to preview the coordinate system used in the work plane compared to the 3D coordinate system.

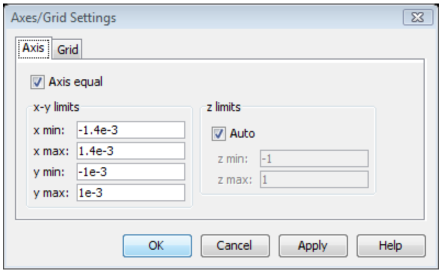
4 Click **OK**.

In order to draw the geometry you must set the size of the drawing table. Do so by returning to the **Options** menu.

Setting Axes and Grid Lines

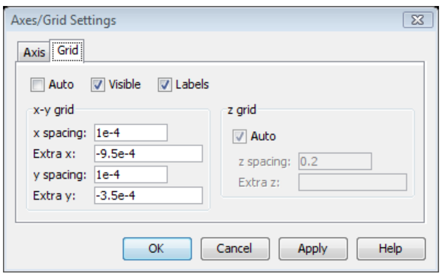
1 From the **Options** menu select **Axes/Grid Settings**.

2 Set the minimum and maximum values of the x and y axes as in the following figure. To do so, click on the edit fields and enter the corresponding values.



3 Click the **Grid** tab, then clear the **Auto** check box.

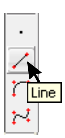
4 Set the **x spacing**, **y spacing**, **Extra x**, and **Extra y** grid lines as in the following figure.



5 Click **OK**.

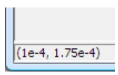
Drawing 2D Objects

You are now ready to draw a cross section of one of the legs of the device soldered to a circuit board. You make extensive use of the Draw toolbar, a vertical toolbar on the left side of the drawing area (see the nearby figure). To find the appropriate button, place the cursor above any toolbar button (without clicking) and its name appears automatically. If you do not know the name of the button referred to in an instruction, take a guess and place the cursor above that button, continuing along the toolbar until you find the correct one.



1 Click the **Line** button on the Draw toolbar.

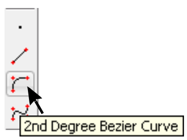
2 In the drawing area, create the first line by clicking the (x, y) coordinates $(-0.8{\cdot}10^{-3}, -0.4{\cdot}10^{-3})$ and then $(-0.4{\cdot}10^{-3}, -0.4{\cdot}10^{-3})$; click the left mouse button on the positions corresponding to the given coordinates. Because you have already placed the grid lines properly, COMSOL Multiphysics snaps to the closest crossing between horizontal and vertical grid lines when you click the mouse near the desired position. Later, if desired, you can deactivate the snap function by double-clicking on **SNAP** in the status bar (the horizontal bar at the bottom of the graphical user interface).



Note: You can read the location of the cursor at any time by looking at the display in the bottom left corner of the COMSOL Multiphysics window.

---

3 Click the **2nd Degree Bézier Curve** button (the one containing an arc) on the Draw toolbar.



4 Create a first arc. On the drawing area, click the coordinate pairs $(0, -4\cdot10^{-4})$ and then $(0, 0)$.

5 Continue using the **2nd Degree Bézier Curve** tool and click on the coordinate pairs $(0, 2\cdot10^{-4})$ and $(2\cdot10^{-4}, 2\cdot10^{-4})$.

6 Again click the **Line** button on the Draw toolbar.

7 Click the coordinate pairs $(8\cdot10^{-4}, 2\cdot10^{-4})$, $(8\cdot10^{-4}4, 4\cdot10^{-4})$, and $(2\cdot10^{-4}, 4\cdot10^{-4})$.

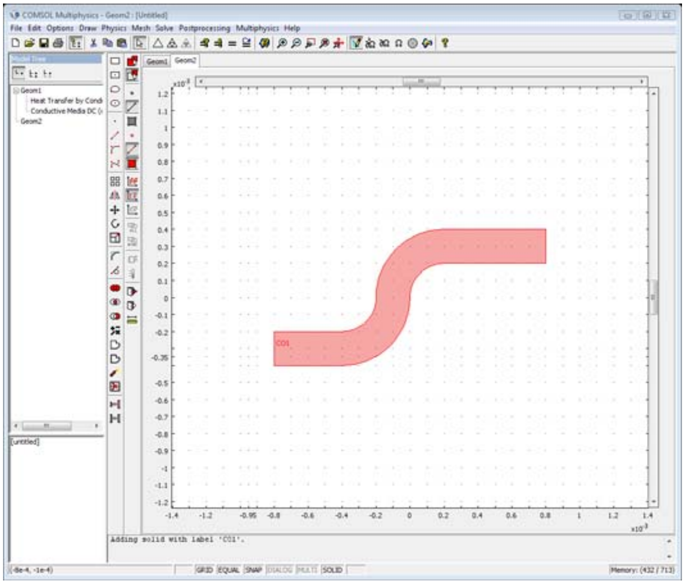
8 Again click the **2nd Degree Bézier Curve** button on the Draw toolbar.

9 Click the coordinate pairs $(-2\cdot10^{-4}, 4\cdot10^{-4})$, $(-2\cdot10^{-4}, 0)$, $(-2\cdot10^{-4}, -2\cdot10^{-4})$, and $(-4\cdot10^{-4}, -2\cdot10^{-4})$.

10 Click the **Line** button on the Draw toolbar.

11 Click the coordinate pair $(-8\cdot10^{-4}, -2\cdot10^{-4})$.

12 To create a solid object that the software labels CO1, click the right mouse button (see the following figure).



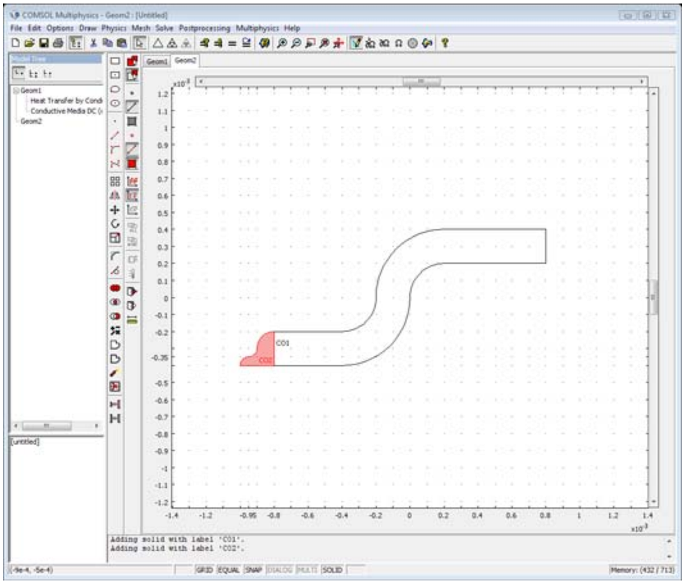
13 Again click the **2nd Degree Bézier Curve** button on the Draw toolbar.

14 Click the coordinate pairs $(-8{\cdot}10^{-4}, -2{\cdot}10^{-4})$, $(-9{\cdot}10^{-4}, -2{\cdot}10^{-4})$, $(-9{\cdot}10^{-4}, -3{\cdot}10^{-4})$, $(-9{\cdot}10^{-4}, -3.5{\cdot}10^{-4})$, $(-9.5{\cdot}10^{-4}, -3.5{\cdot}10^{-4})$, $(-1.0{\cdot}10^{-3}, -3.5{\cdot}10^{-4})$, and $(-1.0{\cdot}10^{-3}, -4{\cdot}10^{-4})$.

15 Click the **Line** button on the Draw toolbar.

16 Click the coordinate pair $(-8{\cdot}10^{-4}, -4{\cdot}10^{-4})$.

17 Click the right mouse button to create a second composite object, CO2.

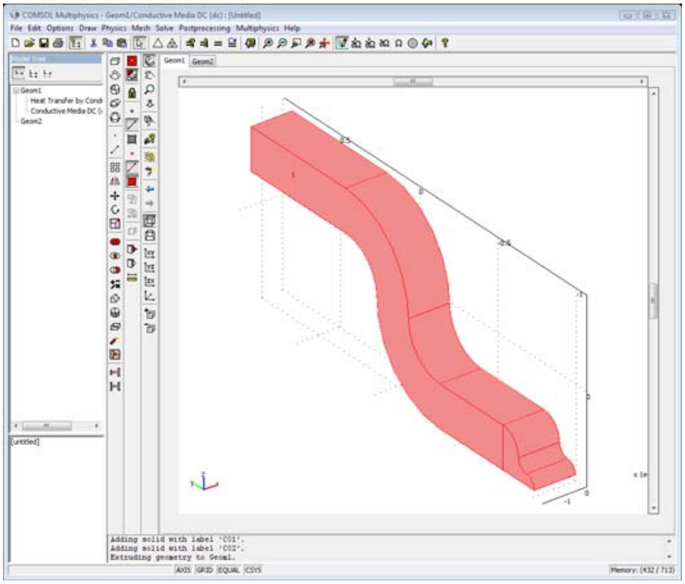


18 In the **Edit** menu choose **Select All**.

Creating 3D Objects With Extrude

1 In the **Draw** menu select **Extrude**. In the **Distance** edit field enter `-0.2e-3` to extrude CO1 and CO2 in a direction perpendicular to the work plane.

2 Click **OK**.



Creating a 3D Object with Revolve

1 Click the **Geom2** tab at the top of the drawing area to go back to the work plane.

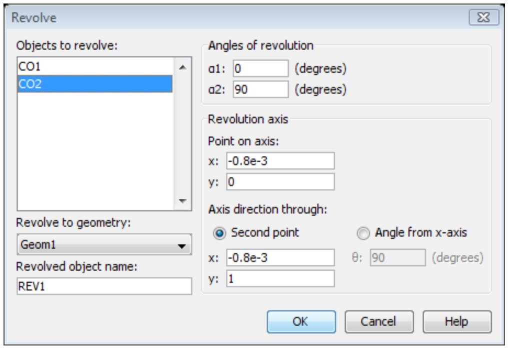
2 Click the object CO2 to highlight it.

3 From the **Draw** menu select **Revolve**.

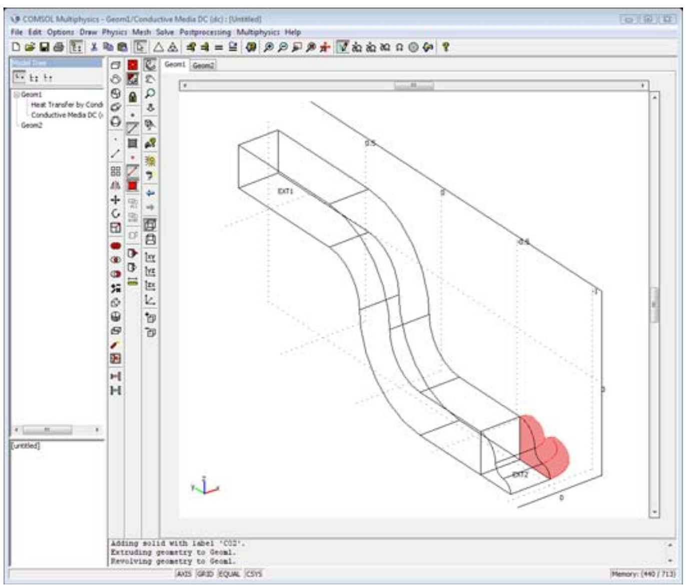
4 Go to the **Angles of revolution** area. In the α**1** edit field enter `0`, and in the α**2** edit field enter `90`.

5 Go to the **Revolution axis** area and then the **Point on axis** fields. In the **x** field enter `-0.8e-3`, and in the **y** field enter `0`. Next locate the **Axis direction through** area and

select the **Second point** option button. In the **x** edit field enter `-0.8e-3`, and in the **y** edit field enter `1`.



6 Click **OK**, and you should see the following geometry:



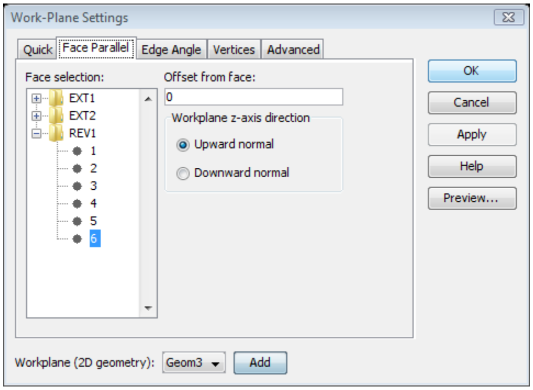
Using Faces to Define Work Planes to Extrude from Existing Surfaces

1 In the **Draw** menu select **Work-Plane Settings**.

2 Click the **Add** button to generate a new work plane in Geom3.

3 Click the **Face Parallel** tab.

4 Click **REV1** to open the folder, then in the **Face selection** list select **6**.



5 Click **OK**.

6 Click the **Zoom Extents** button on the Main toolbar.

7 Click the **Geom2** tab, then select **CO2**.

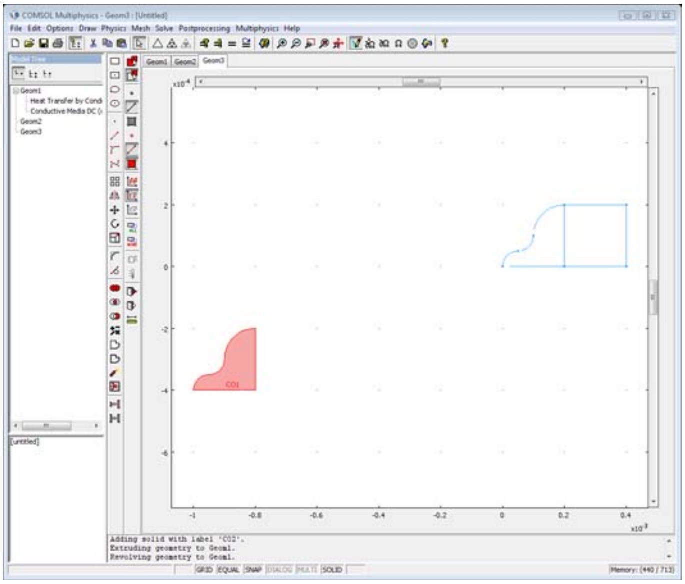
8 From the **Edit** menu select **Copy**.

9 Click the **Geom3** tab.

10 From the **Edit** menu select **Paste**.

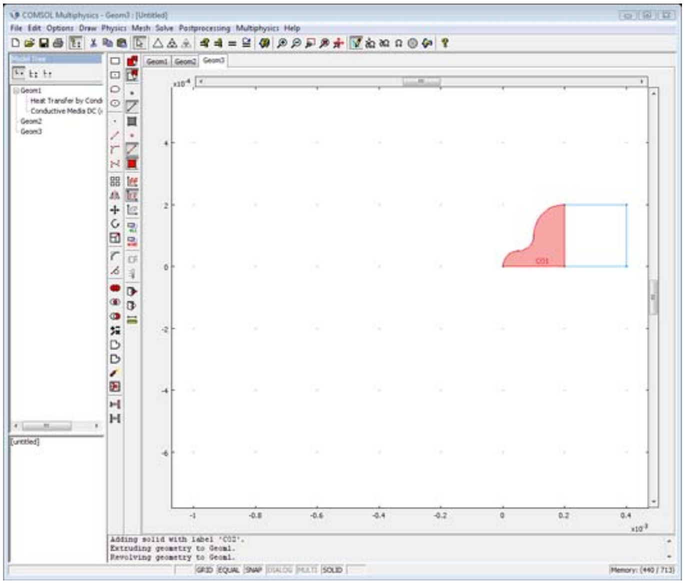
11 In the **Paste** dialog box do not change the entries in the **Displacements** edit fields (the displacements in **x** and **y** directions should be zero), so simply click **OK**.

12 Click the **Zoom Extents** button on the Main toolbar.



The projection of the 3D object on the selected work plane now appears as a blue structure. You must drag and drop CO1 on top of the projection of the corresponding shape.

13 Click and hold the cursor on the right upper corner of CO1, which you drag to the position above the projection of REV1 (see following figure).



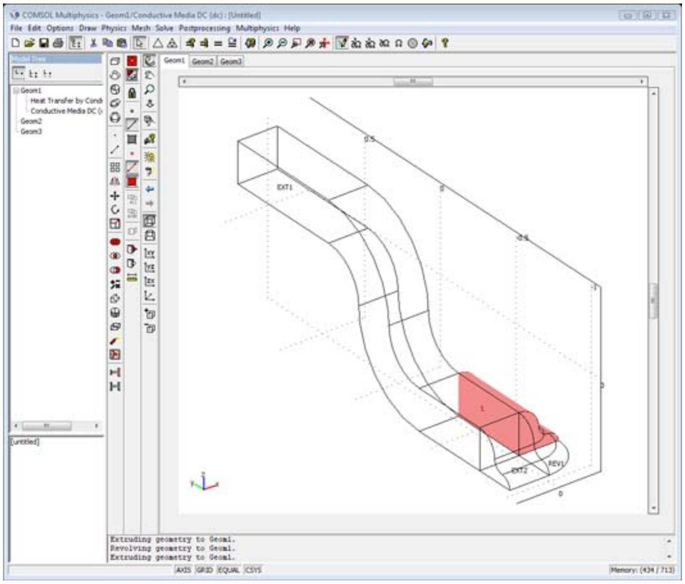
14 Click the **Zoom Extents** button on the Main toolbar. CO1 should still be highlighted, now in red.

15 From the **Draw** menu select **Extrude**.

16 In the **Distance** edit field enter `0.4e-3`.

17 Click **OK**.

You should now see the following object in the 3D drawing table:



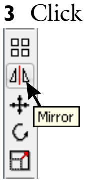
Using Mirror Planes

1 Right click the red object to store the previous selection.

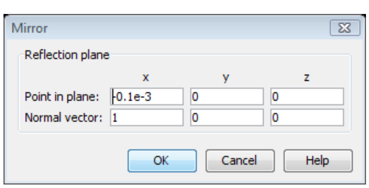
The object should now turn blue, implying that the software is keeping the selection. Use this trick to store an individual selection of an object when you want to make multiple selections of objects from a 3D drawing.

2 Click on top of object REV1. You should now have a blue and red object, REV1 being the red one.

3 Click the **Mirror** button in the Draw toolbar.

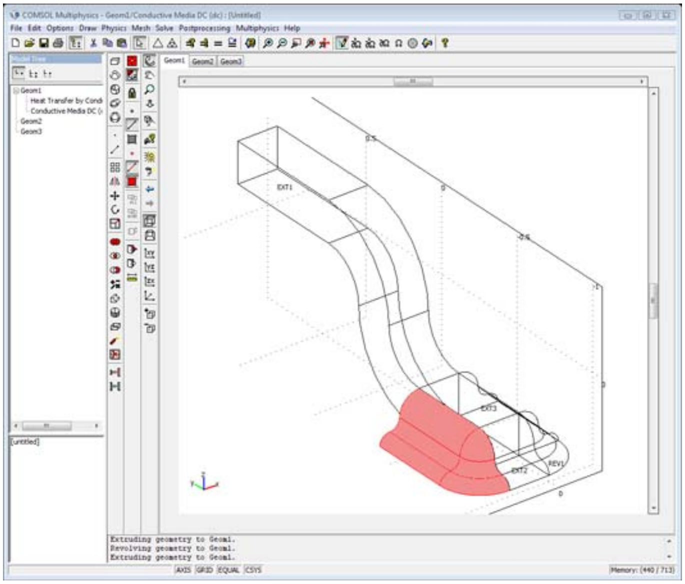


4 Define the **Point in plane** and **Normal vector** using the settings in the following figure:



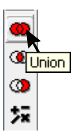
5 Click **OK**.

The drawing area should now resemble this figure:



6 From the **Edit** menu choose **Select All**.

7 Click the **Union** button on the Draw toolbar.



Copy and Paste Using Displacements

1 In the **Edit** menu select **Copy**.

2 In the **Edit** menu select **Paste**.

3 In the **x** edit field enter `1.2e-3`.

4 Click **OK**.

You now have two objects, CO1 and CO3.

5 Click the **Zoom Extents** button on the Main toolbar.

Bridging the Two Legs of the Structure

1 In the **Draw** menu select **Work-Plane Settings**.

2 Click **Add**.

3 Click the **Edge Angle** tab.

4 In the **Edge selection** list select **CO1** and then select edge **36**.

5 Click the **Face selection** button and select face **CO1: 20**.

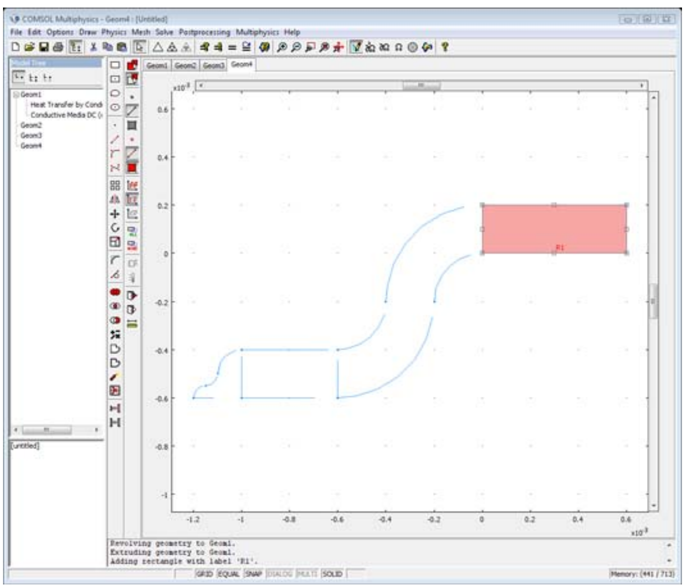
6 Click **Apply** to visualize the position of the work plane in relation to the 3D drawing table.

   You need this information when extruding a profile from the work plane. A negative extrusion generates an object in the space between CO1 and CO3, that is, to the left of the selected face. In order to unite CO1 and CO3 using a block-shaped extrusion, you must extrude in the negative x direction.

7 Click **OK**.

8 Click **Zoom Extents** on the Main toolbar.

9 Click **Rectangle/Square** on the Draw toolbar. Click and hold the mouse button as you drag the cursor from the upper-left corner to the lower-right corner to create the rectangle as in the following figure.

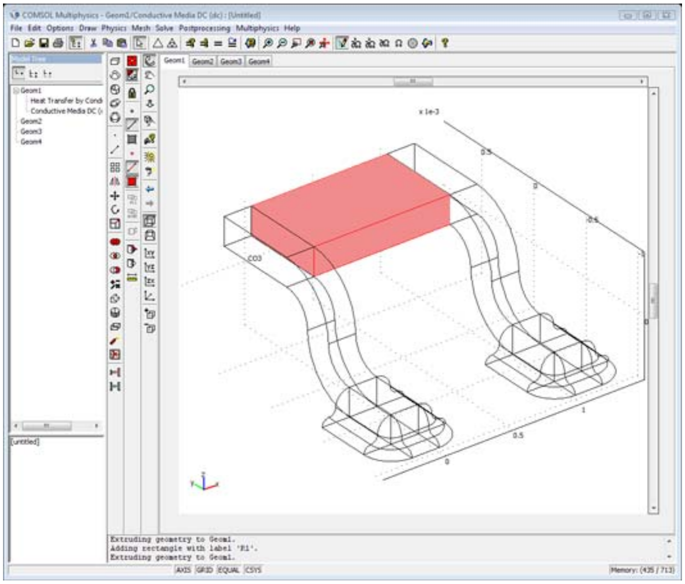


10 From the **Draw** menu select **Extrude**.

11 In the **Distance** edit field enter `-1.0e-3`.

12 Click **OK**.

The drawing area should now resemble the following figure:

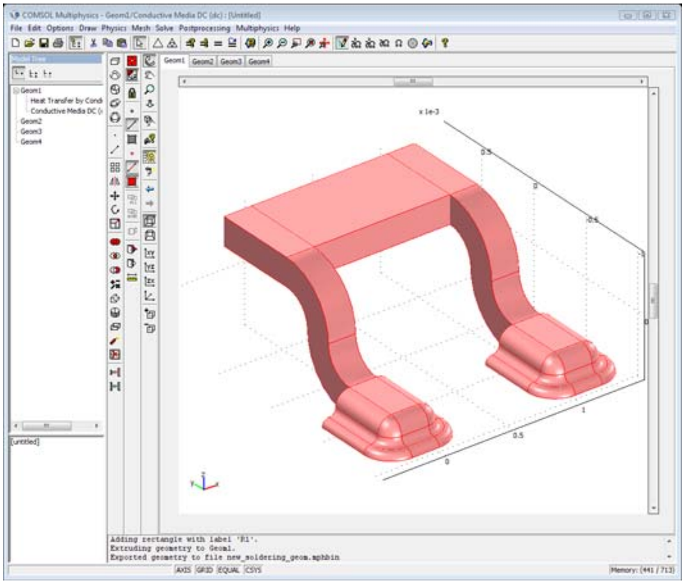


13 From the **Edit** menu choose **Select All**.

14 Click the **Union** button on the Draw toolbar.

Using the Scene Light to Visualize the 3D Object

1 Click the **Scene Light** button on the Camera toolbar (a vertical toolbar on the left side of the user interface) to obtain a better 3D view. The newly created object should resemble the following figure.



You should now save the model so as to continue the modeling session without any danger of losing information.

2 From the **File** menu select **Save**.

3 Select an appropriate folder, and then in the **File name** edit field enter `soldering1`.

4 Click **Save**.

You have now completed the drawing stage of this exercise. Next continue with the physical properties for the object being modeled.

PHYSICS SETTINGS

If you have created the geometry from scratch, continue with **Subdomain Settings** instructions that appear shortly. If you have skipped the geometry section, you must load the model geometry by following these steps:

1 From the **File** menu, select **Import>CAD Data From File**.

2 In the **Import CAD Data From File** dialog box, make sure that the **COMSOL Multiphysics file (*.mphtxt; *.mphbin; ...)** or **All 3D CAD files** is selected in the **Files of type** list.

3 Locate the `new_soldering_geom.mphbin` file (in the same directory as the Model MPH-file `electronic_conductor.mph`), and click **Import**.

4 Click **Zoom Extents** on the Main toolbar.

5 Click the **Scene Light** button on the Camera toolbar.

Subdomain Settings

In this step you describe the device's electrical and thermal properties. In addition, you introduce the heat source created by the ohmic losses in the electric conductors. Note that the model tree gives you a quick access to constants and functions that you might need in the physics settings, without having to open the corresponding menus. You can explore this feature by clicking the **Detail** button in the **Model Tree** area in the graphical user interface.

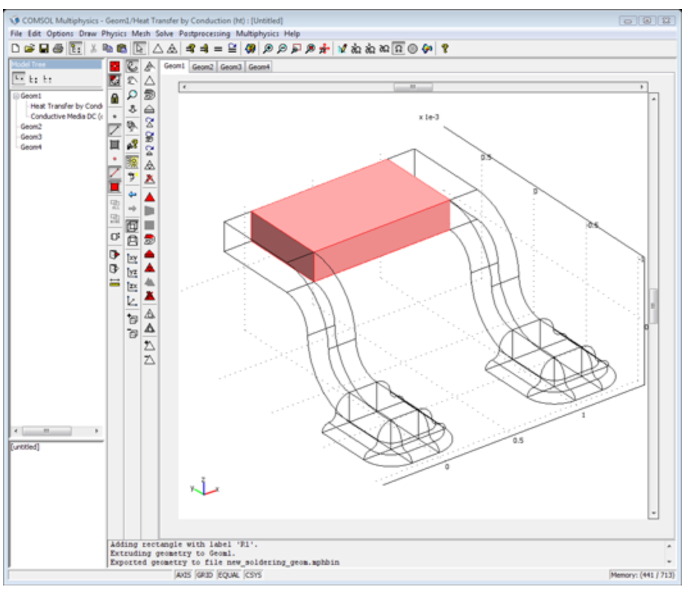
1 From the **Multiphysics** menu, choose **1 Geom1: Heat Transfer by Conduction (ht)**.

2 From the **Physics** menu, choose **Subdomain Settings**.

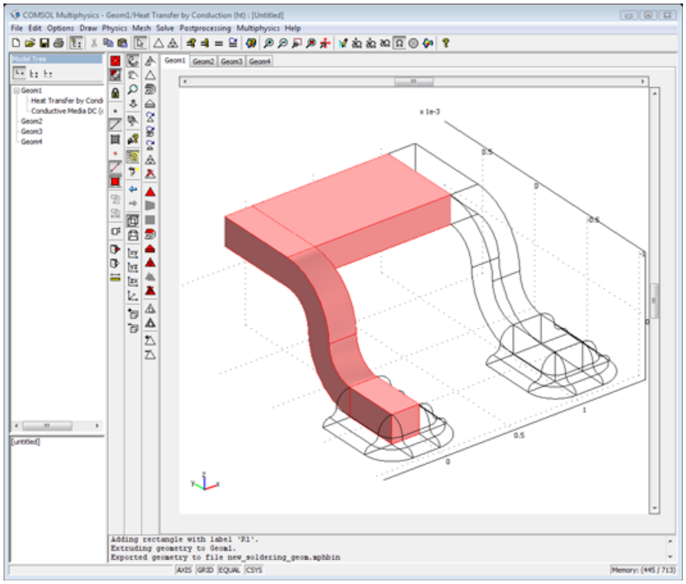
In this multiphysics application, the default procedure is to set the device's thermal properties first and then its electrical properties.

3 Drag the **Subdomain Settings** dialog box away from the main drawing area.

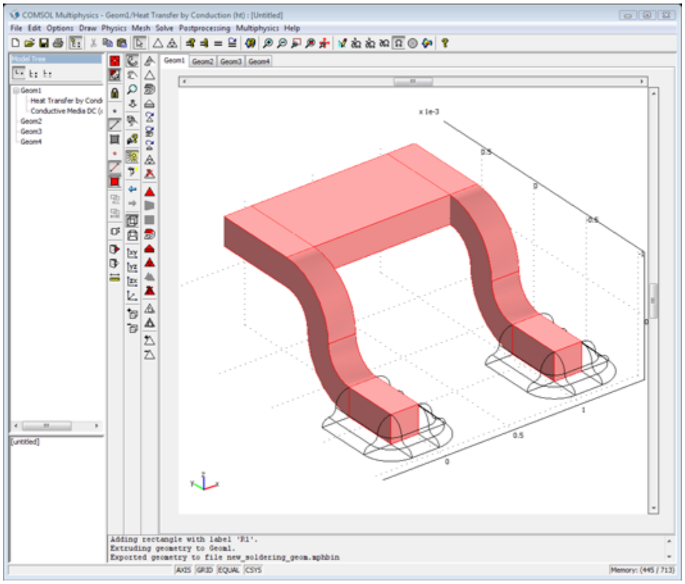
4 In the main graphical user interface click on Subdomain 7 (the one illustrated in the nearby figure) with the left mouse button to highlight it, then click on it again with the right mouse button to store the selection. Subdomain 7 then turns blue.



5 Click on Subdomain 4 (one of the legs, as indicated in the following figure) with the left mouse button to highlight it, then click it again with the right mouse button to store the selection.



6 Click on Subdomain 11 (the other leg as indicated in the following figure) with the left mouse button to highlight it, then click it again with the right mouse button to keep the selection. The drawing area should now resemble the following figure:



7 Go to the **Subdomain Settings** dialog box and click the **Load** button to access the material libraries.

Note that if your license includes the COMSOL Material Library (an add-on product with a library of temperature-dependent material properties), and you have access to a search function. If you use that search function instead of browsing to the **Basic Material Properties** folder, you might select another copper material than the one used in this model. The Basic Material Properties library contains a limited number of materials and is included in COMSOL Multiphysics.

8 In the **Materials** list open the **Basic Material Properties** folder and select **Copper**. Click **OK**.

9 Back in the **Subdomain Settings** dialog box click the **Groups** tab.

10 Locate the **Group selection** area, and in the **Name** edit field enter `connector`.

11 Click the **Subdomains** tab to store the groups.

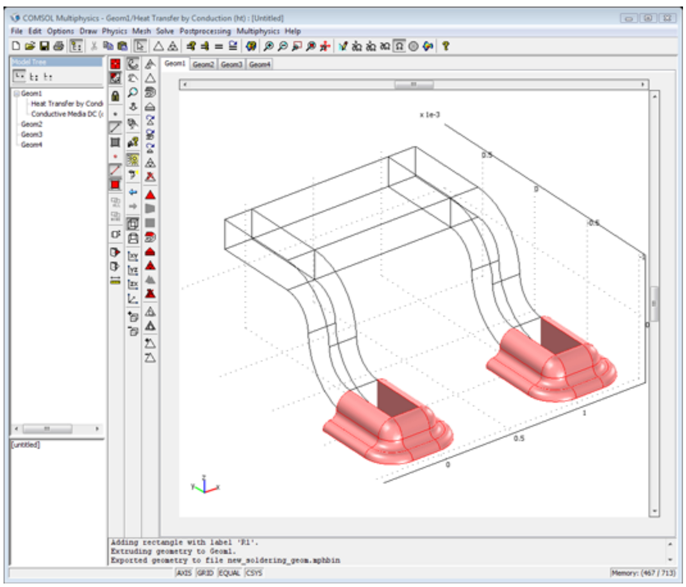
12 Click the **Init** tab, then in the **T(t_0)** edit field enter `Tr`.

13 Click the **Physics** tab.

14 Select the **Select by group** check box.

This action makes it possible to select the subdomains that define the two solder joints with one mouse click because they form the complement to the **connector** group for the device leg. This procedure is not mandatory; you could select each subdomain individually and set its properties. However, the ability to group subdomains according to their properties saves a substantial amount of time during the modeling process if you need to change device properties and experiment with different materials.

15 In the **Subdomain selection** list click on any subdomain numbers that are not highlighted in blue. Doing so should highlight the subdomains corresponding to the solder joints (the complement to the **connector** group) as in the following figure:



Material Properties

Continue with the **Subdomain Settings** operations by defining the material properties for the various subdomains.

1 In the **Subdomain Settings** dialog box click the **Load** button.

2 Go to the **Materials** list, expand **Basic Material Properties**, and select **Solder, 60Sn-40Pb**. Click **OK**.

3 Back in the **Subdomain Settings** dialog box click the **Groups** tab.

4 Locate the **Group selection** area, and in the **Name** edit field delete the string **default** and instead enter `soldering`.

5 Once again click the **Subdomains** tab.

Note that this application mode automatically introduces a heat source denoted **Q_dc**. This source is defined in the both connector (legs) and solder-joint subdomains. You will later investigate the definition of this heat source.

6 Click the **Init** tab, then in the **T(t_0)** edit field enter `Tr`. Click **OK**.

Electrical Properties

You can now continue by setting the electrical properties for the different subdomains.

1 In the **Multiphysics** menu select **2 Geom1: Conductive media DC**.

You can also open the **Subdomain Settings** dialog box from the Model Tree at the left side of the user interface by clicking the **Inspect** toolbar button, open the **Conductive Media DC** folder, and double-click **Subdomain Settings**. Doing so automatically changes the active application mode, so you do not have to do that in the **Multiphysics** menu.

2 In the **Physics** menu select **Subdomain Settings**.

Note that the subdomain grouping from the Heat Transfer application mode is not available in the application mode you have just selected; it is always possible to define different groupings for various application modes.

3 Click the **Load** button. In the **Materials** list open the **Model** folder and select **Solder, 60Sn-40Pb**. Click **OK**.

4 Click the **Groups** tab, then in the **Name** edit field enter `soldering`.

5 Click the **Subdomains** tab, then select the **Select by group** check box.

6 Click the **Init** tab and in the **V(t_0)** edit field enter `Vg`.

7 Click the **Physics** tab.

8 Click any subdomain number not highlighted in the **Subdomain selection** list. Doing so selects the complement to the solder-joints group.

9 Click the **Load** button. In the **Materials** list open the **Model** folder and select **Copper**. Click **OK**.

There are two ways to define a temperature-dependent conductivity. You can use an expression for the conductivity entered directly into the σ edit field, where the temperature enters as a variable. Some of the materials in the material libraries have built-in functions defining the temperature dependence. The other alternative is to use the simple linear relationship for the resistivity mentioned in the introduction of this model. Use this linear relationship for this model.

10 From the **Conductivity relation** list, select **Linear temperature relation**. The Copper material you selected has predefined parameters for this relation. The values of the parameters found for a material show up as bold text in the edit fields.

11 Click the **Groups** tab. Go to the **Name** edit field, delete **default** and enter `connector`.

12 Click the **Subdomains** tab to store the groups.

13 Click the **Init** tab, and in the **V(t_0)** edit field enter `Vg`. Click **OK**.

You have now set all the properties needed to define the physics inside the modeled device. The next step is to specify how the device interacts with the surrounding environment.

Boundary Conditions: Conductive Media DC Application Mode

In the subdomain settings you started by defining the properties for the thermal balance in the Heat Transfer application mode. At this point the Conductive Media DC application mode is active, so it makes sense to set the boundaries for this application mode first and afterwards return to the thermal balance boundary settings.

1 From the **Physics** menu select **Boundary Settings**.

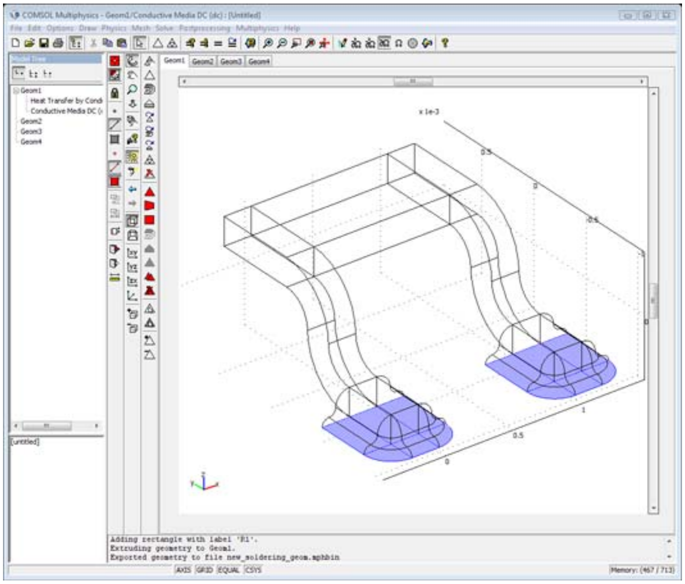
2 In the **Edit** menu choose **Select All**.

3 In the **Boundary condition** list select **Electric insulation**.

4 Go to the main drawing area and click Boundary 2 to first highlight it in red, then right-click to store the selection, which turns the boundary blue.

   Note that selecting faces in 3D is not trivial. Clicking on a point in space might actually select several faces that fall along the cursor's projection. In such cases, continue clicking, whereupon COMSOL Multiphysics toggles between the faces it finds along this projection. If you still cannot manage to select the desired face, move the cursor slightly until you manage to select it. When you have selected the desired face and highlighted it in red, click the right mouse button without moving the cursor to store that selection, so that you can proceed to the next face without clearing the previous selection of faces.

5 Continue selecting faces using the procedure just described until you have the selection as in the following figure:



6 Return to the **Boundary Settings** dialog box, and in the **Boundary condition** list select **Electric potential**.

7 In the **Electric potential** edit field enter `dv`.

Note that you have already defined `dv` in the constants list. You can use the same procedure for expressions, where they can be arbitrary functions of the modeled variables V and T and their derivatives.

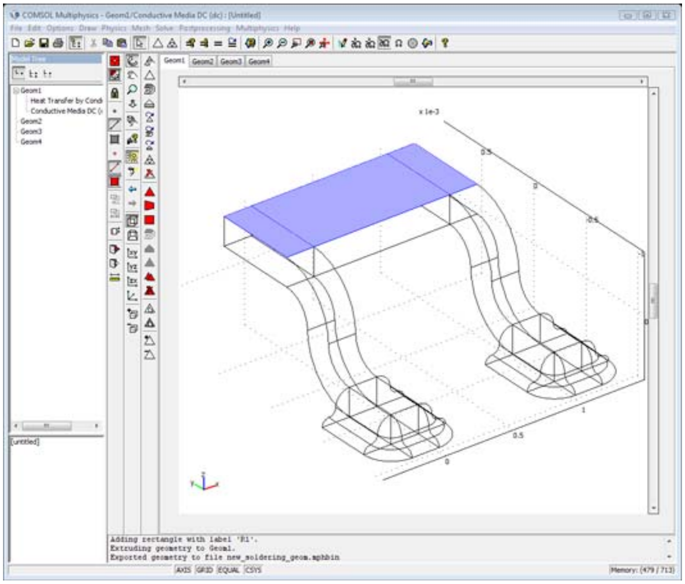
8 Click the **Groups** tab.

9 In the **Name** edit field enter `board`.

10 Click the **Boundaries** tab.

11 From the **Edit** menu choose **Deselect All**.

12 Using the procedure described earlier, select the boundaries in the following figure:



13 Return to the **Boundary Settings** dialog box, and in the **Boundary conditions** list select **Inward current flow**.

14 In the **Normal current density** edit field enter `-kf*(V-Vg)`.

Note that you have already defined `kf`, a fictive film conductance, in the constants list, where `Vg` is also defined. `V` is the dependent variable for the potential in the Conductive Media DC application mode.

15 Click the **Groups** tab.

16 In the **Name** edit field enter `film`.

17 Click the **Boundaries** tab.

18 Select the **Select by group** check box.

19 Click on any boundary number in the **Boundary selection** list that is not denoted `board` or `film` and that is available (that is, does not appear dimmed).

20 Click the **Groups** tab.

21 In the **Name** edit field enter `air`.

22 Click the **Boundaries** tab.

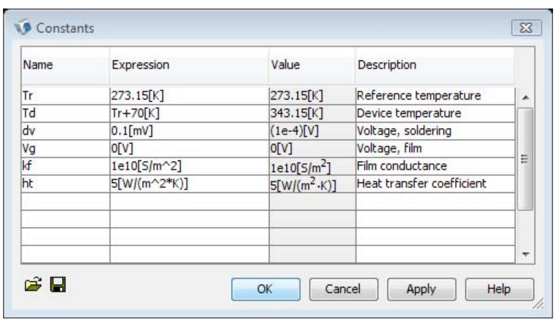
23 Click **OK**.

Boundary Conditions: Heat Transfer Application Mode

You have just defined the boundary conditions for the current balance in the Conductive Media DC application mode. Now continue with the Heat Transfer application mode.

1 In the **Multiphysics** menu select the **Heat Transfer** application mode.
2 From the **Physics** menu select **Boundary Settings**. The software should still have as selected the boundaries corresponding to the **air** boundary group in the electric current balance.
3 In the **Boundary condition** list select **Heat flux**.
4 In the **Heat transfer coefficient** edit field enter `ht`.

   Although you have not yet defined `ht`, you can add and remove constants and expression at any stage of the modeling process. You can now define `ht` while still keeping the **Boundary Settings** dialog box open.
5 From the **Options** menu select **Constants**.
6 In the **Name** column enter `ht`; in the **Expression** column enter `5[W/(m^2*K)]`; and in the **Description** column enter `Heat transfer coefficient` (see the following figure).



7 Click **OK** to close the **Constants** dialog box.
8 Click on the window frame of the **Boundary Settings** dialog box.
9 In the **External temperature** edit field enter `Td`.
10 Click the **Groups** tab.
11 In the **Name** edit field enter `air`.
12 Click the **Boundaries** tab to store the groups.
13 Click **OK**.
14 Click the **Save** button in the Main toolbar before continuing.

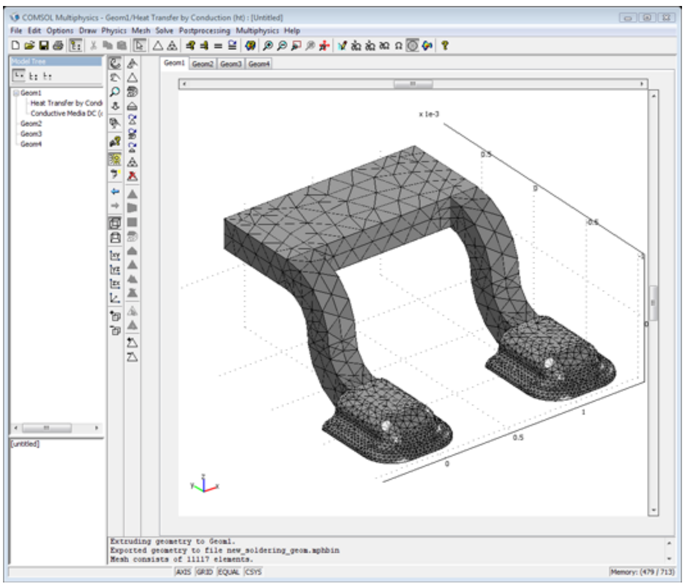
The boundary conditions for the thermal balance are insulating except for those parts exposed to air. This implies that you cannot count on any cooling from other solid parts connected to the device.

You have now completed the model definition, so next proceed to the meshing, solving, and postprocessing steps.

MESH GENERATION

As a first approach, you can use the default mesh-generation settings. Later you can compare the results obtained with this mesh with those using a finer mesh.

1 Click the **Initialize Mesh** button in the Main toolbar. The mesh should resemble that in the following figure.



The default mesh resolves the geometry by creating a denser mesh in areas with high curvature. This is clearly visible in the curved sections of the solder joints, where the mesh is denser than in the parts with a less pronounced curvature.

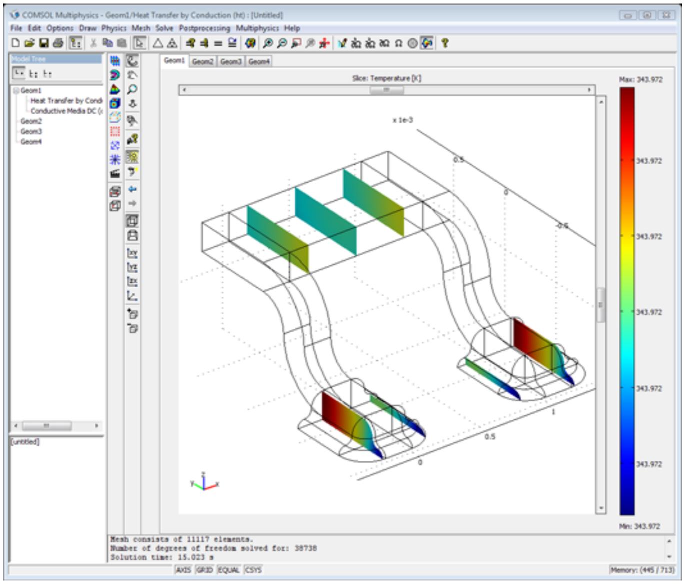
COMPUTING THE SOLUTION

The solver includes both application modes in the solution process by default, taking account for the fully coupled system.

To solve the problem, click the **Solve** button (the one with a "**=**" sign) on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

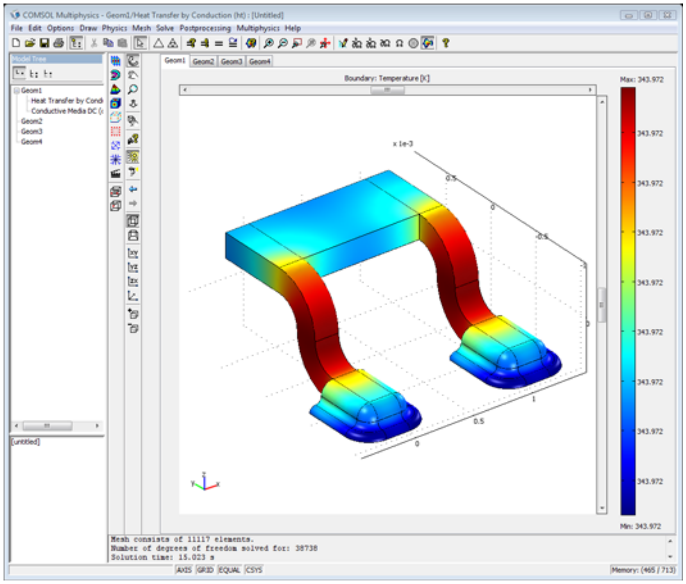
The default result shows a slice plot of the temperature:



Continue by looking at a boundary plot of the temperature distribution, which shows the temperature distribution on the entire boundary:

1 Click the **Plot Parameters** button on the Main toolbar.
2 In the **Plot Parameters** dialog box, click the **General** tab if not already active.
3 In the **Plot type** area select the **Boundary** check box, then clear the **Slice** check box.
4 Click the **Boundary** tab and verify that **Temperature** is selected in the **Predefined quantities** list. If not, select **Heat Transfer by Conduction (ht)>Temperature** from the **Predefined quantities** list.
5 Click **OK**.

6 Click the **Save** button on the Main toolbar.



The resulting plot reveals a very small temperature variation in the device (see the previous figure). You can also see that both the temperature profile is symmetric along a vertical plane between the two legs of the device.

Plotting the electric potential instead (by selecting **Conductive Media DC (dc)> Electric potential** from the **Predefined quantities** list for the plot type that you use) reveals that it is also symmetric along the same plane and the distribution of the ohmic losses in the device (see Figure 2-2 on page 39).

Saving a Model Image

You can now save this model, create a model image, and add some comments regarding the model description. You can later see the model image and read the model properties in the **Model Navigator** when opening COMSOL Multiphysics. This feature is of great use when you have variations of a model in different files, where the individual model files are large and take a long time to load. Reading the model properties and viewing the model image should then guide you to the desired model without having to open it.

1 From the **File** menu choose **Save Model Image**.

2 From the **File** menu choose **Model Properties**.

3 Enter the appropriate description in the different pages by clicking on the respective tabs. The text you type in the **Description** page is then displayed together with the model image when you click the model in the **Model Navigator**. The description and model image appear without you having to open the model.

4 Click **OK**.

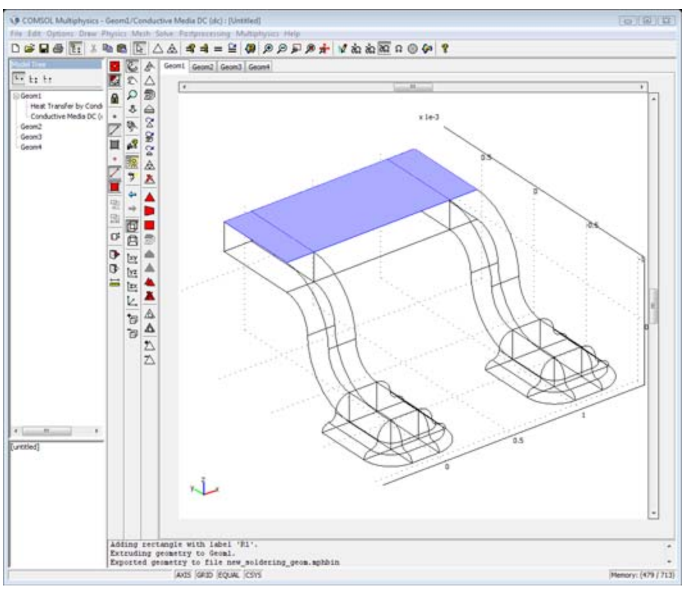
5 Click the **Save** button on the Main toolbar.

Integral Evaluation

In this problem, an interesting analysis is to look at the total current leaving the device at the boundary group denoted **film** in the Conductive Media DC application mode. To do so, go to the **Postprocessing** menu and select **Boundary Integration**, then select the appropriate surface, find the **Expression** edit field, enter `kf*(V-Vg)`, then click **OK**. However, it is necessary to perform this operation every time you run the model. A better alternative is to create an integration-coupling variable to calculate the integral automatically every time you solve the model.

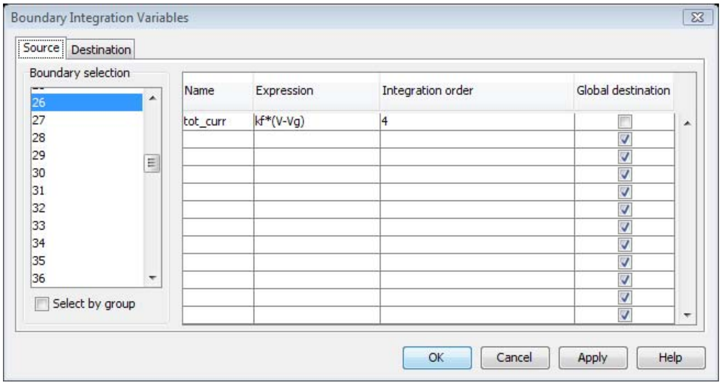
1 Select the menu item **Options>Integration Coupling Variables>Boundary Variables**.

2 Define the selection as in the following figure. As mentioned earlier, you can do so by first clicking with the left mouse button to highlight a surface (it appears in red). Once you have highlighted the desired surface, click the right mouse button to store this selection and continue with the next surface.



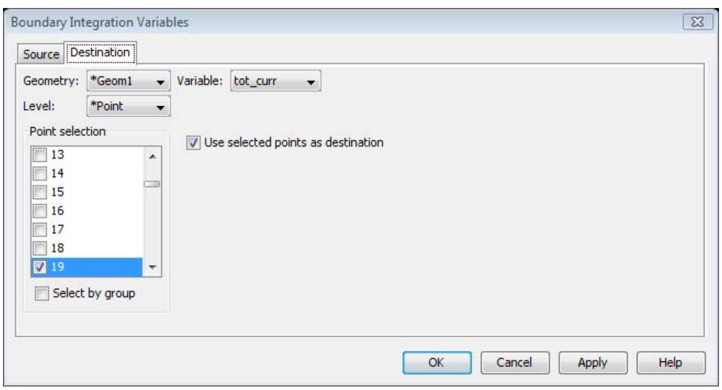
3 In the **Boundary Integration Variables** dialog box go to the **Name** column and enter `tot_curr`. In the **Expression** column enter `kf*(V-Vg)`. Clear the **Global destination** check box.

The dialog box should look like the one in the following figure:



4 Click the **Destination** tab.

5 Store the value of the integral in Point 19 in the **Point selection** list by checking the corresponding check box (see the next figure). Click **OK**.



Note that you can store the value of the integral anywhere. However, it is intuitive to store this value in a point that is associated with the surface where the integral is defined. The model automatically evaluates the integral every time you solve the problem.

You do not need to solve the entire problem again to compute the integral; you simply update the solution to evaluate the integral or any other expression.

6 In the **Solve** menu select **Update Model**.

   You can now display the value of the integral in the message log.

7 In the **Postprocessing** menu select **Point Evaluation**. Select Point 19, if it is not already selected.

8 In the **Expression** edit field enter `tot_curr`. Click **OK**.

The value (approximately 0.24 A) appears in the message log at the bottom of the user interface.

PARAMETRIC ANALYSIS

You have now completed a typical modeling procedure in COMSOL Multiphysics. A natural extension of this model is to run a parametric analysis for different values of the total potential difference, *dv*, across the device and the film boundary. This next section includes the instructions to expand this model to include a parametric analysis.

1 Click the **Solver Parameters** button on the Main toolbar.

2 Go to the **General** page, and in the **Solver** list select **Parametric**.

3 In the **Parameter name** edit field enter `dv`.

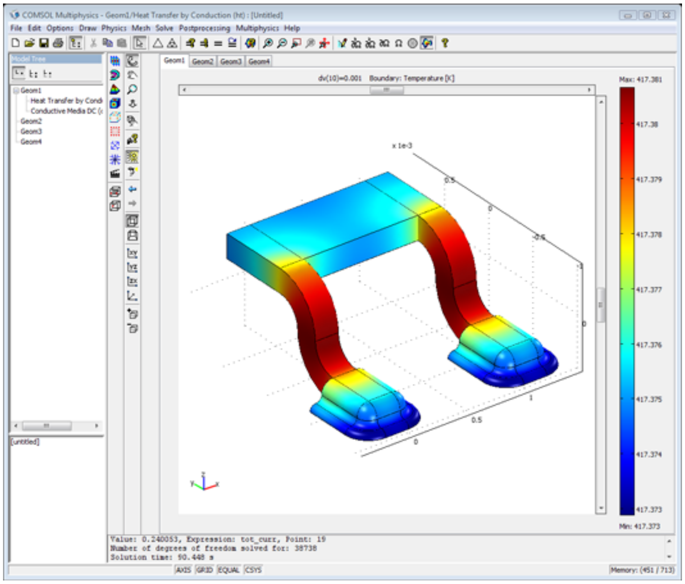
   This step overwrites the value of the constant `dv` in the constant list and replaces it with the parameter value you now define in the corresponding edit field in the next step.

4 In the **Parameter values** edit field enter `1e-4:1e-4:1e-3`.

   You enter this list of parameter values using COMSOL Script syntax, in this case starting with 10^{-4} V and increasing the value with increments of 10^{-4} V up to an ending value of 10^{-3} V.

5 Click **OK**.

6 Click the **Restart** button (the one with a "=" sign with an arrow through it) on the Main toolbar to start with the current solution of the problem at $dv = 10^{-4}$ V. The following figure shows the resulting plot for $dv = 1$ mV.



7 Click the **Save** button on the Main toolbar.

You can now display the total current as a function of dv.

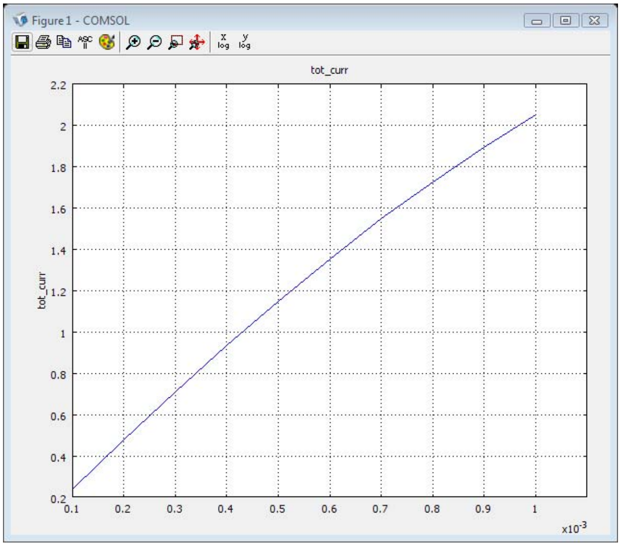
1 In the **Postprocessing** menu select **Domain Plot Parameters**.

2 Click the **Point** tab.

3 In the **Point selection** list select **19** (or select that point directly on the drawing by clicking on it).

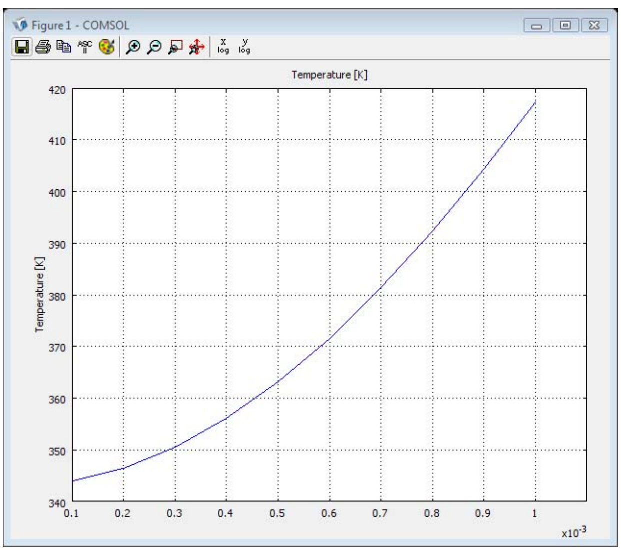
4 In the **Expression** edit field enter `tot_curr`.

5 Click **Apply**. The plot in the following figure should appear in a separate window.



You can also examine the temperature increase at different potentials.

6 In the **Predefined quantities** edit field select **Heat Transfer by Conduction (ht)> Temperature** to create the following plot:



This example provides an introduction to the basic modeling procedure in COMSOL Multiphysics. A proper extension of this model would be to return to the **Mesh** menu and make the mesh either coarser or finer. You can then estimate the mesh convergence by studying the change in total current and temperature for the different cases.

Documenting the Model

To create a report that contains complete information about the model, choose **Generate Report** from the **File** menu. The default settings in the **Generate Report** dialog box provides a report in the HTML format. You can also print the report directly. Select the **Open browser automatically** check box to directly display the report in a web browser. You can configure the contents of the reports by clicking the **Contents** tab and adjust the settings for the report contents. The *COMSOL Multiphysics User's Guide* contains complete documentation of the report generator.

3

Quick Reference

The objective of this *Quick Reference* chapter is to provide a comprehensive list of variables, functions, operators, and equation forms for fast access during modeling. This chapter also lists shortcut keys that you can use to open some of the dialog boxes in the COMSOL Multiphysics user interface.

Equation Forms

Coefficient Form PDE

$$
\begin{cases}
e_a \dfrac{\partial^2 u}{\partial t^2} + d_a \dfrac{\partial u}{\partial t} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f & \text{in } \Omega \\
\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g - h^T \mu & \text{on } \partial\Omega \\
hu = r & \text{on } \partial\Omega
\end{cases}
$$

General Form PDE

$$
\begin{cases}
\nabla \cdot \Gamma = F & \text{in } \Omega \\
-\mathbf{n} \cdot \Gamma = G + \left(\dfrac{\partial R}{\partial u}\right)^T \mu & \text{on } \partial\Omega \\
0 = R & \text{on } \partial\Omega
\end{cases}
$$

Interpreting PDE Coefficients

The COMSOL Multiphysics PDE formulations can model a variety of problems, but note that this documentation uses coefficient names that fall within the realm of continuum mechanics and mass transfer. For the coefficient form:

- e_a is the *mass coefficient*
- d_a is a *damping coefficient* or a *mass coefficient.*
- c is the *diffusion coefficient.*
- α is the *conservative flux convection coefficient.*
- β is the *convection coefficient.*
- a is the *absorption coefficient.*
- γ is the *conservative flux source term.*
- f is the *source term.*

$$\overset{\text{Mass}}{e_a\frac{\partial^2 u}{\partial t^2}} + \overset{\text{Damping/Mass}}{d_a\frac{\partial u}{\partial t}} - \nabla\cdot\underbrace{(\overset{\text{Diffusion}}{c\nabla u} + \overset{\text{Convection}}{\alpha u} - \overset{\text{Source}}{\gamma})}_{\text{Conservative Flux}} + \overset{\text{Convection}}{\beta\cdot\nabla u} + \underset{\text{Absorption}}{au} = \overset{\text{Source}}{f}$$

In some cases, this interpretation does not apply. For instance, a time-harmonic PDE such as the Helmholtz equation represents a time-dependent phenomenon transformed into the frequency domain.

For the Neumann boundary condition of the coefficient form

$$\mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu$$

- q is the *boundary absorption coefficient*.
- g is the *boundary source term*.

Classical PDEs

Many classical PDEs are instances of the coefficient form. All the classical PDEs in this section have their own application modes. To find them, go to the **Model Navigator** and then to the list of application modes; within the **PDE Modes** section find the **Classical PDEs** folder. The table below shows the available classical PDEs using two notations:

the compact notation of vector analysis (used in this documentation) and an expanded mathematical notation.

TABLE 3-1: CLASSICAL PDES IN COMPACT AND STANDARD NOTATION

EQUATION	COMPACT NOTATION	STANDARD NOTATION (2D)
Laplace's equation	$-\nabla\cdot(\nabla u) = 0$	$-\frac{\partial}{\partial x}\frac{\partial u}{\partial x}-\frac{\partial}{\partial y}\frac{\partial u}{\partial y} = 0$
Poisson's equation	$-\nabla\cdot(c\nabla u) = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Helmholtz equation	$-\nabla\cdot(c\nabla u)+au = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au = f$
Heat equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u) = f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Wave equation	$e_a\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u) = f$	$e_a\frac{\partial^2 u}{\partial t^2}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Schrödinger equation	$-\nabla\cdot(c\nabla u)+au = \lambda u$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au = \lambda u$
Convection-diffusion equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+\beta\cdot\nabla u = f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+$ $\beta_x\frac{\partial u}{\partial x}+\beta_y\frac{\partial u}{\partial y} = f$

Mathematical and Logical Functions

The following lists include mathematical and logical functions and operators:

TABLE 3-2: UNARY OPERATORS

OPERATOR	DESCRIPTION
+	unary plus
-	unary minus
~	logical not

TABLE 3-3: 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 3-4: MATHEMATICAL FUNCTIONS AND OPERATORS

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 3-4: MATHEMATICAL FUNCTIONS AND OPERATORS

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`
`exp`	exponential	`exp(x)`
`flc1hs`	smoothed Heaviside function	`flc1hs(x,scale)`
`flc2hs`	smoothed Heaviside function	`flc2hs(x,scale)`
`flsmhs`	smoothed Heaviside function	`flsmhs(x,scale)`
`flsmsign`	smoothed sign function	`flsmsign(x,scale)`
`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)`

TABLE 3-4: MATHEMATICAL FUNCTIONS AND OPERATORS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
`max`	maximum of two arguments	`max(a,b)`
`min`	minimum of two arguments	`min(a,b)`
`mod`	modulo operator	`mod(a,b)`
`NaN, nan`	not-a-number	`nan`
`real`	real part	`real(u)`
`pi`	pi	`pi`
`sec`	secant	`sec(x)`
`sech`	hyperbolic secant	`sech(x)`
`sign`	sign 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)`

TABLE 3-5: VECTOR-CREATION FUNCTIONS AND OPERATORS

FUNCTION	DESCRIPTION
`linspace(`*`start`*`, `*`stop`*`, `*`N`*`)`	linearly spaced vector
`logspace(`*`start`*`, `*`stop`*`, `*`N`*`)`	logarithmically spaced vector
`:`	vector creation operator

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.
- 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.

Variables

Geometry Variables

In the tables below, x (in italic font) indicates the name of any space coordinate in the current model, for example, x, y, and z. u indicates the name of a dependent variable. Replace it with the actual names of the dependent variables in your model, for example, T for temperature.

TABLE 3-6: GEOMETRY VARIABLES

VARIABLE NAME	DESCRIPTION
`x y z`	Default space coordinate names, Cartesian coordinates
`r phi z`	Default space coordinate names, cylindrical coordinates
x`g`	Space coordinate values of the original geometry
`s`	Curve parameter in 2D (0 to 1 in direction of the boundary arrow)
`s1 s2`	Arc length parameters in 3D
`t`*x*	Curve tangent vector, x component (2D)
`t1`*x* `t2`*x*	Surface tangent vectors, x component (3D)
`n`*x*	Outward unit normal vector, x component
`dn`*x*	Down direction normal vector, x component
`un`*x*	Up direction normal vector, x component
`h`	Mesh element diameter
`dom`	Domain number
`dvol`	Determinant of the Jacobian relating local space coordinate values to the global coordinate values

Field Variables

Table summarizes the field variables that are available in all COMSOL Multiphysics models. The table does not include *application mode variables*, which vary depending on the application modes in your model. See the *COMSOL Multiphysics Modeling*

Guide and the module documentation for more information about available application mode variables.

TABLE 3-7: FIELD VARIABLES

VARIABLE NAME	DESCRIPTION
u	Dependent variable
ux	Dependent variable, space derivative w.r.t x
ux_ix_j	Dependent variable, second space derivative w.r.t x_i and x_j
ut	Dependent variable, first time derivative
$u$tt	Dependent variable, second time derivative
ux_it	Dependent variable, mixed space and first time derivative
ux_itt	Dependent variable, mixed space and second time derivative
uTx	Tangential derivative variable

Miscellaneous Variables

TABLE 3-8: MISCELLANEOUS VARIABLES

VARIABLE NAME	DESCRIPTION
`t`	time
`lambda`	eigenvalue (for postprocessing only)
`phase`	phase factor

Operators

TABLE 3-9: OPERATORS

OPERATOR	DESCRIPTION
`diff(`*f*`, `*x*`)`	Differentiation operator. Differentiation of *f* with respect to *x*.
`pdiff(`*f*`, `*x*`)`	Differentiation operator. Differentiation of *f* with respect to *x*. No chain rule for dependent variables.
`test(`*expr*`)`	Test function operator
`nojac(`*expr*`)`	No contribution to the Jacobian
`up(`*expr*`)`	Evaluate expression as defined in adjacent up side
`down(`*expr*`)`	Evaluate expression as defined in adjacent down side
`mean(`*expr*`)`	Mean value of expression as evaluated on adjacent boundaries
`depends(`*expr*`)`	True if expression depends on the solution
`islinear(`*expr*`)`	True if expression is a linear function of the solution
`dest(`*expr*`)`	Evaluate parts of an integration coupling expression on destination side.
`if(`*cond*`,`*expr1*`,`*expr2*`)`	Conditional expression evaluating the second or third argument depending on the value of the condition
`quad(`*f*`,`*x*`,`*a*`,`*b*`,`*tol*`)`	Adaptive numerical quadrature of *f* with respect to *x*
`with(`*solnum*`,`*expr*`)`	Access any solution during postprocessing
`at(`*time*`,`*expr*`)`	Access the solution at any time during postprocessing

Shortcut Keys

Pressing the shortcut keys in the following table opens a dialog box.

TABLE 3-10: SHORTCUT KEYS

SHORTCUT KEY	DIALOG BOX
F1	Help Desk
F5	Point Settings
F6	Edge Settings
F7	Boundary Settings
F8	Subdomain Settings
Ctrl+F5	Equation System>Point Settings
Ctrl+F6	Equation System>Edge Settings
Ctrl+F7	Equation System>Boundary Settings
Ctrl+F8	Equation System>Subdomain Settings
F9	Free Mesh Parameters
Ctrl+F9	Mapped Mesh Parameter
F11	Solver Parameters
F12	Plot Parameters

9

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 *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 any 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.

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

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu & \text{on } \partial\Omega \\ hu = r & \text{on } \partial\Omega \end{cases}$$

coerce To convert a *geometry object* from one type to another, for example, from a *curve object* to s 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 `.mphtxt` 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 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 B*ézier* and *NURBS* curves and surfaces are defined by a set of points known as *control points*. The locations of the 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:

$$\lambda^2 e_a u - \lambda d_a u + \nabla \cdot (-c\nabla u - \alpha u) + \beta \cdot \nabla u + au = 0$$

elliptic PDE A linear stationary 2nd-order elliptic PDE has the form

$$\nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f$$

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 structure 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

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot\Gamma = F & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = G + \left(\dfrac{\partial R}{\partial u}\right)^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

generalized Neumann boundary condition A generalized Neumann boundary condition (also called a *mixed boundary condition* or a *Robin boundary condition*) specifies the value of a linear combination of the *normal flux* and the *dependent variables* on a boundary. For a coefficient form PDE, the generalized Neumann boundary condition is

$$\mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu$$

The generalized Neumann condition is often called just *Neumann condition* in the documentation.

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*

$$e_a\frac{\partial^2 u}{\partial t^2} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f$$

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 tetrahedrons (3D). These are examples of *mesh elements*. See also *grid*, *structured mesh* and *unstructured mesh*.

mesh element The individual elements in the mesh that together form a partitioning of the geometry, for example, *triangular elements* and *tetrahedral elements*. See also *finite element*.

mesh 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*

$$d_a\frac{\partial u}{\partial t} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f$$

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 a*pplication 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

$$\dot{x} = Ax + Bu$$
$$\dot{y} = Cx + Du$$

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*.

INDEX