COMSOL MULTIPHYSICS®

Model Library

VERSION 3.4



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: +39-030-3793890 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 I 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 AL 10 9AB Phone:+44-(0)-1707 284747 Fax: +44-(0)-1707 284746 info.uk@comsol.com www.uk.comsol.com

United States

COMSOL, Inc. I 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 Model Library © COPYRIGHT 1994–2007 by COMSOL AB. All rights reserved

Patent pending

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

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

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

Version: October 2007 COMSOL 3.4

CONTENTS

Chapter I: Introduction

Typographical Conventions .	•	•	·	•	•	•	·	•	•	•	•	·	•	•	•	•	•	•	3
Model Library Guide																			5

Chapter 2: Acoustics Models

Acoustics of a Muffler													10
Introduction										•	•		10
Model Definition										•	•		10
Results and Discussion				•	•	•		•					П
Modeling in COMSOL Multiphysics					•	•							12
Modeling Using the Graphical User Interface	•	•	•	•	•	•	•	•	•	·	•	•	13
Eigenmodes of a Room													19
Model Definition													19
Results and Discussion													21
Modeling Using the Graphical User Interface													22

Chapter 3: Chemical Engineering Models

Transport and Adsorption	30
Model Definition	31
Modeling in COMSOL Multiphysics	33
Results	34
Modeling Using the Graphical User Interface	36
Tubular Reactor	42
Introduction	42
Model Definition	42
Results	45

Modeling Using the Graphical User Interface	•	•		•	•	•	•	•	47
0 0 1									

Chapter 4: Electromagnetics Models

Electrostatic Potential Between Cylinde	rs											55
Model Definition												55
Results												55
Modeling Using the Graphical User Interface												56
Pacemaker Electrode												66
Introduction					•			•		•	•	66
Model Definition										•		67
Results and Discussion			•				•					68
Modeling Using the Graphical User Interface			•	•	•	•	•			•		69
Skin Effect in Circular Wire												74
Model Definition										•		74
Results				•		•				•		76
Modeling Using the Graphical User Interface												76
Modeling a Wire with an Aluminum Core .			•				•					80
Modeling Using the Programming Language .			•	•	•		•			•		81
Quadrupole Lens												83
Introduction		•	•		•		•	•	•	·	•	83
Model Definition		•	•	•	•	•	•			•		83
Results				•		•				•		86
Modeling Using the Graphical User Interface			•	•		•	•			•		88
Spherical Capacitor												93
Introduction		•	•	•	•		•			•		93
Model Definition										•		94
Modeling Using the Graphical User Interface												95

Chapter 5: Equation-Based Models

Electrical Signals in a Heart								100
Background				•		•		100
Model Definition				•				102
Modeling in COMSOL Multiphysics				•		•		103
Results				•		•		104
Modeling Using the Graphical User Interface				•		•		105
References		•		•	•	•	•	110
A Transport Problem								ш
Background				•		•		111
Model Definition								111
Streamline Diffusion								112
Upwinding Using Discontinuous Basis Functions .								112
Results								113
Modeling in COMSOL Multiphysics								114
Modeling Using the Graphical User Interface								114
Modeling Using COMSOL Script		•		•		•	•	117
An Integro-Partial Differential Equation								119
An Integro-Partial Differential Equation								119 119
An Integro-Partial Differential Equation		•			•	•		119 119 120
An Integro-Partial Differential Equation Introduction	 		 			•		119 119 120 121
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model	· · · ·		 					119 119 120 121 122
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference	· · · ·		 	· ·				119 120 121 122 123
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface	· · · · · ·		· · · · · ·				· · ·	 119 120 121 122 123 124
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results. Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons	· · ·	•	 	• •			· · ·	 119 120 121 122 123 124 128
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction	· · ·		· · ·	· • • • • •			· · · · · ·	 119 120 121 122 123 124 128 128
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction Model Definition	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · ·	· • • • • • • • • • • • • • • • • • • •	· · · ·	· · ·	· · · · · · · · ·	 119 120 121 122 123 124 128 129
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction Model Definition Results.	· · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	 119 120 121 122 123 124 128 128 129 130
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction Model Definition Results. Results.	· · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	 119 119 120 121 122 123 124 128 129 130 130
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction Model Definition Results References Model Definition Model Definition References Modeling Using the Graphical User Interface	· · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·		· · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	 119 119 120 121 122 123 124 128 129 130 131
An Integro-Partial Differential Equation Introduction Modeling in COMSOL Multiphysics Results. Comparison with the Full 3D Radiation Model Reference Modeling Using the Graphical User Interface The KdV Equation and Solitons Introduction Model Definition References Model Definition References Model Definition The KdV Equation and Solitons Introduction Model Definition Results. References Modeling Using the Graphical User Interface The Telegraph Equation	· · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · ·		· · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	 119 119 120 121 122 123 124 128 128 129 130 130 131 135

Model Definition					•			•	·		•	135
Results												136
Modeling in COMSOL Multiphysics					•						•	138
Modeling Using the Graphical User Interface	•	•	•	•	•	•	•		•	•	•	139
Shallow Water Equations												142
Introduction					•						•	142
Artificial Stabilization												143
Model Definition												143
Results and Discussion												146
Modeling in COMSOL Multiphysics												147
Reference												147
Modeling Using the Graphical User Interface	•			•	•	•		•	•	•	•	147
Shell Diffusion												150
Introduction												150
Model Definition												150
Results												151
Modeling in COMSOL Multiphysics												153
Modeling Using the Graphical User Interface	•								•			153
Spherically Symmetric Transport												156
												156
Introduction		•	•	•	•	•	•	•		•	•	
Introduction	•	•	•	•	•	•	•	•		•		156
Introduction	•		•	•	•	•	•	• •	•	•	•	156 159
Introduction												156 159 161
Introduction												156 159 161 162
Introduction												156 159 161 162 166
Introduction		· · ·		· · ·	· · ·	156 159 161 162 166 166						
Introduction	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	156 159 161 162 166 166 167
Introduction	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · ·	· · · ·	· · · ·	· · · · · · · · ·	· · ·	· · ·	156 159 161 162 166 166 167 167
Introduction	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · ·	156 159 161 162 166 166 167 167
Introduction	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	156 159 161 162 166 166 167 167 168 169
Introduction		· · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	156 159 161 162 166 167 167 167 168 169 171
Introduction	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · · · · · · ·	· · · ·	· · · ·	156 159 161 162 166 166 167 167 168 169 171						
Introduction	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	156 159 161 162 166 167 167 167 168 169 171 171 173						

Results and Discussion									175
Modeling in COMSOL Multiphysics			•					•	178
References	•			•	•	•			179
Modeling Using the Graphical User Interface	•		•					•	179

Chapter 6: Fluid Dynamics Models

Backstep with Argyris Element	192
Model Definition	192
Argyris Element Properties	192
Results	193
Using the Graphical User Interface	193
Flow Past a Cylinder	198
Model Definition	198
Results	200
Reference	200
Modeling Using the Graphical User Interface	201
Time-Dependent Simulation	204
Fluid Valve	208
Introduction	208
Model Definition	208
Results	210
Modeling in COMSOL Multiphysics	213
Modeling Using the Graphical User Interface	213
Micromixer	219
Introduction	219
Model Definition	220
Results	221
Modeling Using the Graphical User Interface	221
Shock Tube	230
Introduction	230
Model Definition	230

Modeling in COMSOL Multiphysics	231
Results	232
Modeling Using the Graphical User Interface	233
Terminal Falling Velocity of a Sand Grain	237
Introduction	237
Model Definition	238
Results	239
Reference	243
Modeling Using the Graphical User Interface	243
Sloshing Tank	247
Introduction	247
Model Definition	248
Results	250
Modeling Using the Graphical User Interface	25 I

Chapter 7: Geophysics Models

Groundwater Flow and Solute Transport	258
Introduction	258
Model Definition	258
Results	262
References	265
Modeling Using the Graphical User Interface	265
A Rock Fracture Flow Model	270
Introduction	270
Model Definition	270
Modeling in COMSOL Multiphysics	271
Modeling Using the Graphical User Interface	272

Chapter 8: Heat Transfer Models

Heating with a Moving Laser						278
Model Definitions		•				279
Results and Discussion					•	282
Modeling Using the Graphical User Interface		•				284
Heat-Sink Experiments Using the Component Libra	ry					289
Heat-Sink Experiments Using the Component Libra	ry					289 289
Heat-Sink Experiments Using the Component Libra Introduction	.ry	•	•	•	•	289 289 289
Heat-Sink Experiments Using the Component Libra Introduction		•	•	•		289 289 289 292

Chapter 9: Multidisciplinary Models

Magnetic Brake	302
Introduction	302
Model Definition	302
Modeling in COMSOL Multiphysics	304
Results	304
Modeling Using the Graphical User Interface—Fixed ω	305
Modeling Using the Programming Language—Fixed ω	309
Exporting to Simulink—Time-Dependent ω	310
Modeling Using the Graphical User Interface —Time-Dependent ω_{-} .	313
Process Control Using a PID Controller	317
Process Control Using a PID Controller	3 7 3 7
Process Control Using a PID Controller	3 7 3 7 3 9
Process Control Using a PID Controller Image: Control Using a PID Controller Model Definition Image: Control Using a PID Controller Results Image: Control Using a PID Controller Modeling Using the Graphical User Interface Image: Control Using a PID Controller	317 317 319 322
Process Control Using a PID Controller Image: Control Using a PID Controller Model Definition Image: Controller Modeling Using the Graphical User Interface Image: Controlling Temperature	3 7 3 7 3 9 322 322 329
Process Control Using a PID Controller Image: Control Using a PID Controller Model Definition Image: Controlling Temperature Introduction Image: Controlling Temperature	3 7 3 7 3 9 322 322 329
Process Control Using a PID Controller Image: State Stat	3 7 3 7 3 9 3 22 3 29 3 29 3 29
Process Control Using a PID Controller : Model Definition . : Results. : Modeling Using the Graphical User Interface : Controlling Temperature : Introduction : Model Definition : Modeling in COMSOL Multiphysics :	317 319 322 329 329 329 329 330

Using the Graphical User Interface—No Controller					333
Exporting to Simulink—Temperature Control					336

Chapter 10: Multiphysics Models

Magnetic Drug Targeting in Cancer Therapy	342
Introduction	342
Model Definition	342
Model Data	346
Results	346
Modeling in COMSOL Multiphysics	349
References	350
Modeling Using the Graphical User Interface	350
Free Convection	357
Introduction	357
Model Definition	357
Modeling in COMSOL Multiphysics	359
Results	360
Modeling Using the Graphical User Interface	361
Time-Dependent Simulation	373
Modeling Using the Graphical User Interface	373
Marangoni Convection	377
Introduction	377
Model Definition	377
Modeling in COMSOL Multiphysics	379
Results	380
Reference	382
Modeling Using the Graphical User Interface	382
Simulation of a Microrobot	389
Introduction	389
Model Definition	389
Results and Discussion	392
Modeling in COMSOL Multiphysics	393

Reference	•	•								•		393
Modeling Using the Graphical User Interface	•		•	•	•	•	•	•	•	•		393
Vibrations in Milk Containers												399
Model Definition												399
Modeling in COMSOL Multiphysics												400
Results	•											40 I
Modeling Using the Graphical User Interface	•	•	•	•	•	•	•	•	•	•	•	402
Peristaltic Pump												408
Introduction								•				408
Model Definition												408
Results										•		410
Modeling in COMSOL Multiphysics										•		414
Modeling Using the Graphical User Interface										•		414

Chapter II: Quantum Mechanics Models

The Schrödinger Equation for the Hydrogen Atom 422	2
Introduction	2
Model Definition	.2
Results	3
Modeling in COMSOL Multiphysics	5
Modeling Using the Graphical User Interface	8
Conical Quantum Dots 432	2
Conical Quantum Dots 432 Introduction 43	2 2
Conical Quantum Dots 432 Introduction	2 2 2
Conical Quantum Dots 432 Introduction	2 2 5
Conical Quantum Dots 432 Introduction	2 2 5 5
Conical Quantum Dots 432 Introduction	2 2 5 5 6

Chapter 12: Semiconductor Device Models

Semiconductor Diode	442
Introduction	442
Model Definition	444
Results	448
References	45 I
Modeling Using the Graphical User Interface	45 I
Distributed SPICE Model of an Integrated Bipolar Transistor	457
Introduction	457
Model Definition	458
Results and Discussion	459
Modeling in COMSOL Multiphysics	46 I
Reference	46 I
Modeling Using the Graphical User Interface	462
DC Characteristics of a MOS Transistor (MOSFET)	474
Introduction	474
Model Definition	475
Results and Discussion	479
Modeling Using the Graphical User Interface	48 I

Chapter 13: Structural Mechanics Models

Eigenvalue Analysis of a Crankshaft	496
Model Definition	496
Results	496
Modeling Using the Graphical User Interface	499
Deformation of a Feeder Clamp	503
Deformation of a Feeder Clamp	503 503
Deformation of a Feeder Clamp Introduction Model Definition	503 503 503
Deformation of a Feeder Clamp Introduction	503 503 503 505

Stresses in a Pulley	522
Introduction	522
Model Definition	522
Results and Discussion	524
Modeling in COMSOL Multiphysics	526
Modeling Using the Graphical User Interface	527
A Note About Adaptive Mesh Refinement	535
Stiffness Analysis of a Communication Mast's Diagonal Mounting	537
Introduction	537
Model Definition	538
Results	541
Modeling using the Graphical User Interface	542
3D Analysis of a Formula One Car Front Wing	549
Introduction	549
Model Definition	549
Results	553
Reference	554
Modeling Using the Graphical User Interface	554

Chapter 14: Wave Propagation Models

Diffraction Patterns	560
Introduction	560
Model Definition	560
Results	561
Modeling Using the Graphical User Interface	561
Modeling Using the Programming Language	564

Chapter 15: Benchmark Models

Isospectral Drums					568
Introduction				•	568
Model Definition				•	568

Results	•			•	•		•		569
Modeling in COMSOL Multiphysics	•				•		•		570
References	•				•		•		570
Modeling Using the Graphical User Interface								•	570
Modeling Using the Programming Language .	•			•	•	•	•		573

K
C

577

Introduction

We have designed COMSOL Multiphysics to be an extremely flexible package applicable to virtually any area of research, science, and engineering. A consequence of this flexibility is that it becomes necessary to set up COMSOL Multiphysics for a specific modeling task. To relieve you of this work in many disciplines and application areas, we have created a series of time-saving preconfigured models and collected them in this *Model Library*. It ships as a standard component of COMSOL Multiphysics 3.4, and it is just one of several collections available to COMSOL Multiphysics users. Similar libraries with models devoted to specific disciplines are also available in the range of add-on modules:

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

These ready-to-run models cover many classic problems and equations from science and engineering, and they have two goals in mind:

- To show the versatility of COMSOL Multiphysics and the wide range of applications it covers
- To form an educational basis for you to learn about COMSOL Multiphysics and gain an understanding of the underlying physics

Each model entry consists of two elements: the documentation and a corresponding ready-to-run COMSOL Multiphysics Model MPH-file you load directly from the Model Navigator window in COMSOL Multiphysics. These model files are copied to your hard disk with the default installation.

The documentation for each model entry in this library includes a technical background, the model definition, and results. Each entry also contains a section that shows, step by step, how to implement the model in COMSOL Multiphysics. Following the model description, a series of instructions guide you in the process of building, solving, and postprocessing the model in COMSOL Multiphysics.

Note: In some cases, the settings appear in a table rather than giving a step-by-step procedure for each data entry.

Those users new to COMSOL Multiphysics might want start reading the *COMSOL Multiphysics Quick Start and Quick Reference*, which contains a detailed step-by-step instruction for a basic multiphysics model. During modeling, some general questions might arise about COMSOL Multiphysics, its various features, and how to use them. For the answers we suggest that you first refer to the COMSOL Multiphysics User's Guide and the COMSOL Multiphysics Modeling Guide.

The Model Library entries follow the basic structure of COMSOL Multiphysics and the modeling process you follow in using this package. Most of the chapters correspond to *application modes* in COMSOL Multiphysics. For a description of these application modes, including typical example models included in the Model Library, see the *COMSOL Multiphysics Modeling Guide*.

When describing *equation-based models*, which use partial differential equations (PDEs) directly, the text provides information about the model and the physical or mathematical background. In *benchmark models* you can compare a solution to analytical or established results. *Parametric studies* vary physical or geometrical

parameters in a series of solutions for optimization or evaluation of design criteria. To quickly locate models in each of the above categories, refer to the "Model Library Guide" on page 5.

One outstanding features of COMSOL Multiphysics is the ability to include several interacting physics in one system using its multiphysics modeling features. To help you better appreciate this power, the Model Library includes a number of corresponding examples in the chapter "Multiphysics Models" on page 341. Additional information about multiphysics modeling is available in the *COMSOL Multiphysics User's Guide* and the *COMSOL Multiphysics Modeling Guide*.

The section "Multidisciplinary Models" on page 301 contain models that include control loops or other system simulation components. The models in this chapter show you how to combine COMSOL with MATLAB and Simulink for system simulations that include finite element models.

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.

Model Library Guide

The table below summarizes key information about the entries in this Model Library.

The first three columns contain the model name, page number, and solution times. The solution time is the elapsed time measured on a machine running Windows Vista with a 2.6 GHz AMD Athlon X2 Dual Core 500 CPU and 2 GB of RAM. For models with a sequential solution strategy, the Solution Time column shows the elapsed time for the longest solution step.

The subsequent three columns indicate whether the model geometry includes 1D, 2D, 3D, or multiple geometries (extended multiphysics).

The next several columns indicate the analysis types that the model covers, such as stationary, time-dependent, eigenvalue, or nonlinear.

The Multiphysics column shows which models include multiphysics couplings.

The final column indicates models that use parametric studies to evaluate and optimize their results by varying the value of one or more parameters.

MODEL	PAGE	TION TIME				ONARY	DEPENDENT	VALUE	INEAR	IPHYSICS	METRIC STUDY
		.nhos	₽	2D	ñ	STATI	TIME-I	EIGEN	NON	MULT	PARAI
ACOUSTICS MODELS	9										
automotive_muffler	10	31 min			\checkmark	\checkmark					\checkmark
eigenmodes_of_room	19	12 s			\checkmark			\checkmark			
reactive_muffler*	34	4 s		\checkmark		\checkmark					
BENCHMARK MODELS	567										
isospectral_drum l	568	ls		\checkmark				\checkmark			
isospectral_drum2	568	ls		\checkmark				\checkmark			
lshaped_membrane_parametric*	425	ls		\checkmark				\checkmark			
minimal_surface*	296	ls		\checkmark		\checkmark			\checkmark		
point_source*	279	ls		\checkmark		\checkmark					
poisson_unit_disk*	275	ls		\checkmark		\checkmark					

TABLE I-I: COMSOL MULTIPHYSICS MODEL LIBRARY MODELS

TABLE I-I: COMSOL MULTIPHYSICS MODEL LIBRARY MODELS

MODEL	PAGE										٧
		Ψ					ENT			N	STUI
		I NO				VARY	PEND	VLUE	EAR	Ηλεία	TRIC
		ŬĽŊ				TIO	E-DE	ENV	NLIN	LTIP	RAME
		sol	₽	50	ã	ST/	Σ F	BIB	° z	Ρ	PAF
CHEMICAL ENGINEERING MODELS	29										
adsorption	30	5 s		\checkmark		\checkmark				\checkmark	
tubular_reactor	42	30 s		\checkmark		\checkmark			\checkmark	\checkmark	
DIFFUSION MODELS											
effective_diffusivity*	56	5 s		\checkmark			\checkmark				
thin_layer_diffusion*	380	ls			\checkmark	\checkmark					
ELECTROMAGNETICS MODELS	53										
electric_sensor*	118	2 s		\checkmark		\checkmark					
electrochemical_polishing*	414	4 s		\checkmark			\checkmark		\checkmark		
pacemaker_electrode	66	5 s			\checkmark	\checkmark					
permanent_magnet*	123	3 s		\checkmark		\checkmark					
potential_between_cylinders	55	ls			\checkmark	\checkmark					
quadrupole	83	ls		\checkmark		\checkmark					
skin_effect	74	ls		\checkmark		\checkmark					
spherical_capacitor	93	ls		\checkmark		\checkmark					
thin_film_resistance*	368	3 s			\checkmark	\checkmark					
EQUATION-BASED MODELS	99										
black-scholes_put	166	ls	\checkmark								
heart_clg	100	8 min			\checkmark		\checkmark			\checkmark	
heart_fhn	100	32 min			\checkmark		\checkmark			\checkmark	
integro_partial	119	3 s	\checkmark				\checkmark				
kdv_equation	128	10 s	\checkmark				\checkmark		\checkmark		
shallow_water	142	ls	\checkmark				\checkmark				
shell_diffusion	150	ls			\checkmark	\checkmark					
spherical_symmetry_ore	156	ls	\checkmark								
telegraph_equation	135	ls	\checkmark				\checkmark				
transport_problem	111	ls		\checkmark		\checkmark					

MODEL	PAGE										Z
		SOLUTION TIME	ē	ZD	30	STATIONARY	TIME-DEPENDENT	EIGENVALUE	NONLINEAR	MULTIPHYSICS	PARAMETRIC STUI
two_term_boltzmann	171	3 min				\checkmark					\checkmark
FLUID DYNAMICS MODELS	191										
backstep*	150	16 s		\checkmark		\checkmark			\checkmark		\checkmark
backstep_argyris	192	25 s		\checkmark		\checkmark			\checkmark		
backstep_quad*	150	42 s		\checkmark		\checkmark			\checkmark		\checkmark
cylinder_flow	198	9 min		\checkmark			\checkmark		\checkmark		
falling_sand	237	2 min		\checkmark			\checkmark		\checkmark		
fluid_valve	208	22 min		\checkmark			\checkmark				
micromixer	219	31 min			\checkmark	\checkmark			\checkmark		
shock_tube	230	8 s		\checkmark		\checkmark			\checkmark		
sloshing_tank	247	2 min		\checkmark			\checkmark				
GEOPHYSICS MODELS	257										
groundwater_flow	258	7 s		\checkmark			\checkmark			\checkmark	
rock_fracture	270	4 s		\checkmark		\checkmark					
HEAT TRANSFER MODELS	277										
heat_convection_2D*	184	l s		\checkmark		\checkmark					
heat_radiation_ID*	180	l s	\checkmark			\checkmark			\checkmark		
heat_transient_axi*	188	l s		\checkmark			\checkmark				
laser_heating	278	2 min			\checkmark		\checkmark			\checkmark	
lpd19_dev_merge	289	3 min			\checkmark		\checkmark				\checkmark
MULTIDISCIPLINARY MODELS	301										
magnet_brake	302	5 s		\checkmark		\checkmark	\checkmark				
magnet_brake_simulink	302	l s		\checkmark		\checkmark					
PID_control	317	2 min		\checkmark			\checkmark		\checkmark	\checkmark	
thermal_controller_simulink	329	2 s								\checkmark	
MULTIPHYSICS MODELS	341										
electronic_conductor***	36	4 min			\checkmark	\checkmark			\checkmark	\checkmark	

TABLE I-I: COMSOL MULTIPHYSICS MODEL LIBRARY MODELS

_

TABLE I-I: COMSOL MULTIPHYSICS MODEL LIBRARY MODELS						
MODEL	PAGE					
		Ξ				

MODEL	PAGE										ž
		SOLUTION TIME	<u>0</u>	2D	3D	STATIONARY	TIME-DEPENDENT	EIGENVALUE	NONLINEAR	MULTIPHYSICS	PARAMETRIC STUD
free_convection	357	13 s		\checkmark		\checkmark			\checkmark		
magnetic_drug_targeting	342	2 min		\checkmark			\checkmark		\checkmark	\checkmark	
marangoni	377	14 s		\checkmark		\checkmark			\checkmark		
microrobot	389	50 s		\checkmark			\checkmark				
milk_container	399	ls		\checkmark		\checkmark	\checkmark				
peristaltic_pump	408	5 min		\checkmark					\checkmark		
resistive_heating*	334	4 s		\checkmark		\checkmark	\checkmark		\checkmark		
QUANTUM MECHANICS MODELS	421										
conical_quantum_dot	432	ls		\checkmark				\checkmark			
hydrogen_atom	422	3 s		\checkmark				\checkmark			
SEMICONDUCTOR DEVICE MODELS	441										
bipolar_transistor	457	2 min		\checkmark		\checkmark			\checkmark		
mos_transistor	474	4 min		\checkmark		\checkmark			\checkmark		\checkmark
semiconductor_diode	442	2 min		\checkmark		\checkmark			\checkmark		\checkmark
STRUCTURAL MECHANICS MODELS	495										
crankshaft	495	2 min			\checkmark			\checkmark			
edge_load_2d*	226	ls		\checkmark		\checkmark					
feeder_clamp	503	9 s			\checkmark	\checkmark					
fl_front_wing	549	II min			\checkmark	\checkmark					
gravity_load_2d*	231	ls		\checkmark		\checkmark					
mast_diagonal_loading	537	29 s			\checkmark	\checkmark					
pulley	522	10 s		\checkmark		\checkmark					\checkmark
WAVE PROPAGATION MODELS	559										
diffraction_patterns	560	ls		\checkmark		\checkmark					

* In the COMSOL Multiphysics Modeling Guide

** In the COMSOL Multiphysics User's Guide

*** In the COMSOL Multiphysics Quick Start and Quick Reference

Acoustics Models

This chapter contains acoustics models solving for the pressure field using time-harmonic and eigenvalue formulation in the Acoustics application mode.

Acoustics of a Muffler

Introduction

This model describes the pressure wave propagation in a muffler for an explosion engine. The approach is general for analysis of damping of propagation of harmonic pressure waves.

The purpose of the model is to show how to treat 3D acoustics in a fairly complex geometry, consisting of several separate sections and pipes divided by thin perfectly rigid walls. The analysis gives the transmission loss in the frequency range 100–1000 Hz.

Model Definition

The model geometry consists of three separate resonator chambers divided by thin walls. The inlet and the outlet correspond to the connection in the direction of the engine and of free air, respectively.



Figure 2-1: The geometry of a muffler. The exhaust fumes enter through the left pipe, pass the three resonator chambers, and exit through the right pipe.

DOMAIN EQUATIONS

The problem is solved in the frequency domain using the time-harmonic Acoustics application mode. The model equation is a slightly modified Helmholtz's equation for the acoustic pressure, *p*:

$$\nabla \cdot \left(-\frac{\nabla p}{\rho}\right) - \frac{\omega^2 p}{c_s^2 \rho} = 0$$

where ρ is the density, c_s is the speed of sound, and ω is the angular frequency. The density needs to be included in the equation in cases where variations in density in different materials exist. The model assumes that in the low-frequency range, reactive damping prevails. Resistive damping is therefore not included.

BOUNDARY CONDITIONS

The boundary conditions are of three different types. At all the solid boundaries, which include the outer walls of the muffler, the dividing walls between the resonator chambers, and the walls of the pipes, sound hard (wall) boundary conditions are used:

$$\left(-\frac{\nabla p}{\rho}\right) \cdot \mathbf{n} = 0 \tag{2-1}$$

At the inlet boundary is a combination of incoming and outgoing plane waves:

$$\left(-\frac{\nabla p}{\rho}\right) \cdot \mathbf{n} = \frac{i\omega}{\rho c_s} p - \frac{2i\omega}{\rho c_s} p_0$$

In this equation p_0 denotes the applied outer pressure and *i* the imaginary unit. At the outlet boundary, an outgoing plane wave is set:

$$\left(-\frac{\nabla p}{\rho}\right) \cdot \mathbf{n} = \frac{i\omega}{\rho c_s} p$$

Results and Discussion

The following equation defines the damping of the acoustic energy, d_w :

$$d_w = 10 \log\left(\frac{w_o}{w_i}\right)$$

Here, w_0 and w_i denote the acoustic energy at the outlet and inlet, respectively. The acoustic energy is calculated using the following equations:

$$w_{0} = \int_{\partial \Omega} \frac{\left| p_{c} \right|^{2}}{2\rho c_{s}} dA$$

$$w_{i} = \int_{\partial \Omega} \frac{p_{0}^{2}}{2\rho c_{s}} dA$$

Figure 2-2 shows the result of a parametric frequency study. This plot reveals that the damping is better at higher frequencies, with the exception of several deep dips throughout the frequency range. The dips correspond to the resonance frequencies for different parts of the muffler system.



Figure 2-2: The damping (dB) in the muffler as a function of the frequency (Hz).

Modeling in COMSOL Multiphysics

The dividing walls between the resonator chambers and the walls of the pipes constitute interior boundaries. Because no waves propagate through these boundaries, the pressure between the inside and the outside of the pipe walls, as well as that between the chambers, must be decoupled. To accomplish this, use three dependent variables for the pressure (and three Acoustics application modes) in the model: one for the pipes, p_p , one for the first and the third resonator chambers, p_{c1} , and one for the second resonator chamber, p_{c2} . You implement this by deactivating the variables in the parts of the model where they do not exist. It is then possible to specify the dividing walls and the walls of the pipes as exterior boundaries for each pressure

variable. The pressure variables couple to each other where the pipes open up into the resonator chambers and at the openings between the chambers. You only have to specify this coupling in one direction; the implementation distributes the reaction forces evenly. To view the results for the entire pressure field simultaneously, you define a variable p for the common pressure.

Model Library path: COMSOL_Multiphysics/Acoustics/automotive_muffler

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, click the Multiphysics button.
- 2 Add three Acoustics application modes. To do so, select **3D** in the **Space dimension** list and then select **Acoustics>Acoustics>Time-harmonic analysis** in the list of application modes (in the **COMSOL Multiphysics** folder if the license contains additional modules).
- **3** Before adding each application mode, name it and its dependent variable according to the following table:

APPLICATION MODE NAME	DEPENDENT VARIABLE
pipe	p_p
chamber1	p_c1
chamber2	p_c2

Type the names of the application mode and the dependent variable in the **Application mode name** and **Dependent variables** edit fields, respectively.

Application mode name and Dependent variables cuit noids, respective

- 4 Click Add to add each of the application modes to the model.
- 5 When you have added the three application modes, click OK to exit the Model Navigator and continue modeling.

OPTIONS AND SETTINGS

- I Open the **Constants** dialog box from the **Options** menu, and enter the variable name p0 and the expression 1. Click **OK**.
- 2 From the Physics menu, choose Scalar Variables to open theApplication Scalar Variables dialog box. Set the frequency to freq for all the three

application modes. It is sufficient to type freq in the **Expression** edit field for the variable freq_pipe only and press Enter. The synchronization of equivalent variables then sets the two other frequency variables to freq. When done, click **OK**.

GEOMETRY MODELING

- Create a cylinder with radius 0.03, height 0.75, axis base point (-0.1, 0, 0), and axis direction vector (1, 0, 0).
- 2 Create another cylinder with radius 0.03, height 0.65, axis base point (0.25, -0.09, 0), and axis direction vector (1, 0, 0).
- **3** Go to the **Work-Plane Settings** dialog box in the **Draw** menu. On the **Quick** page, specify a **y-z** plane at x = 0. Click **OK**.
- **4** In this new work plane, specify an ellipse with semiaxes 0.15 and 0.07, centered at (0, 0).
- **5** In the **Extrude** dialog box in the **Draw** menu, select the ellipse and extrude it by a distance of 0.8.
- 6 In the Work-Plane Settings dialog box, set the x coordinate for the y-z plane to 0.25. When done, click OK.
- 7 Specify a circle with radius 0.03, centered at (0.09, 0).
- **8** Go to the **Embed** dialog box in the **Draw** menu. Mark both the circle and the ellipse, and embed them.
- **9** Go to the **Work-Plane Settings** dialog box again, and set the **x** coordinate for the **y-z** plane to **0.65**. Click **OK**.
- **IO** Embed the circle and the ellipse from this plane.
- II Finally, click the Zoom Extents button to view the complete muffler geometry.

The geometry should now look like that in Figure 2-3.



Figure 2-3: The muffler geometry.

PHYSICS SETTINGS

Subdomain Settings

In this model, use the default values for the subdomain settings. However, every subdomain should have only one active application mode. Open the **Subdomain Settings** dialog box from each application mode, select the **Active in this mode** check box in the following subdomains, and clear it in all other subdomains.

APPLICATION MODE	ACTIVE IN SUBDOMAINS
ріре	I, 3, 5, 6, 8, 9
chamber I	2, 7
chamber2	4

Boundary Conditions

I In the pipe application mode, select Radiation condition from the Boundary condition list and then select Plane wave from the Wave type list on Boundaries 1 and 50; the pressure source p_p0 should be p0 on Boundary 1 and 0 on Boundary 50. Use the default sound-hard boundary condition on all other boundaries.

- 2 In the chamber1 application mode, use the Pressure boundary condition on Boundaries 19, 31, 40, and 43. Set the pressure source p_cl₀ to p_p on Boundaries 19 and 40 and p_c2 on boundaries 31 and 43. Use the default sound-hard boundary condition on all other active boundaries.
- **3** In the **chamber2** application mode, use the default sound-hard boundary condition on all boundaries.

Coupling Variables

- I On the **Options** menu, point to **Integration Coupling Variables** and then click **Boundary Variables**.
- 2 In the Boundary Integration Variables dialog box, select Boundary 1 and create a boundary integration variable with Name Iin, Expression p0^2/(2*1.25*343), Integration order 4, and Global destination.
- Select Boundary 50 and create a boundary integration variable with Name Iout, Expression p_p^2/(2*1.25*343), Integration order 4, and Global destination.

Expression Variables

- I Open the Scalar Expressions dialog box from the Options menu. Create a scalar expression with Name dw and Expression 10*log10(Iin/Iout). This represents the damping in dB.
- **2** Open the **Boundary Expressions** dialog box. Create a boundary expression with **Name** p and **Expressions** according to the following table:

EXPRESSION	BOUNDARIES
P_P	2–5, 10, 11, 13, 14, 20–23, 25, 26, 28, 29, 36–39, 46–49
p_cl	6, 7, 12, 16, 33, 41, 44
p_c2	17, 27, 32

3 Open the **Subdomain Expressions** dialog box. Create a subdomain expression with **Name** p and **Expressions** according to the following table:

EXPRESSION	SUBDOMAINS
P_P	I, 3, 5, 6, 8, 9
p_cl	2, 7
p_c2	4

MESH GENERATION

Click the Initialize Mesh toolbar button to generate the mesh.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box.
- 2 Select the Parametric solver from the Solver list.
- **3** Type freq in the **Parameter name** edit field and 100:10:1000 in the **Parameter values** edit field. This will compute the solution for 91 equally spaced frequencies from 100 to 1000 Hz. Note that this process takes around 25 minutes; if you want to run a faster analysis, try the same frequency range but with a step of 100 Hz instead (to do so, type 100:100:1000).
- 4 Click OK.
- 5 Click the Solve button to compute the solution.

POSTPROCESSING AND VISUALIZATION

- I To see what goes on inside the muffler, hide the boundaries that obstruct the view. To do so, choose Options>Suppress Suppress Boundaries and suppress boundaries 1, 8, 9, 15, 18, 19, 24, 30, 31, 34, 35, 40, 42, 43, 45, and 50 by selecting them from the Boundary selection list in the Suppress Boundaries dialog box and then click OK.
- 2 Open the Plot Parameters dialog box.
- **3** Visualize the pressure in the muffler with a boundary plot of the absolute value of the pressure and an isosurface plot of the pressure. To do so, click the **General** tab, clear the **Slice** check box and select the **Boundary** and **Isosurface** check boxes in the **Plot type** area.
- 4 Click the Boundary tab and type abs(p) in the Expression edit field.
- 5 Click the Isosurface tab and type p in the Expression edit field in the Isosurface data area. The suitable number of isosurface levels for the isosurface plot varies with the frequency. At frequencies with low damping many of the isosurfaces tend to congregate inside the pipe. For a frequency of 490 Hz (select this solution from the General page of the Plot Parameters dialog box), ten isolevels gives a nice plot.
- 6 Click OK.
- 7 Click the Headlight toolbar button to make the visualization more viewer friendly. The plot should look like the one in Figure 2-4.

8 Click the **Go to XY View** toolbar button to see a projection of the plot on the *xy*-plane. You may need to click it twice, in case the muffler appears upside down. The pressure in the second chamber displays a resonance pattern.



Figure 2-4: The solution at 490 Hz. The real value of the pressure is plotted as isosurfaces, and the absolute value of the pressure is displayed as a boundary plot on the inner walls of the muffler.

- **9** To see the damping as a function of the frequency, open the **Cross-Section Plot Parameters** dialog box.
- 10 On the General page, select all solutions in the Solutions to use area.
- II Click the **Point plot** button.
- 12 Click the Point tab and type dw in the Expression edit field to make a point plot of the damping in some arbitrary point located inside the geometry. The origin (the default point) will do.

The plot should look like in Figure 2-2. Notice the deep dip in the damping around 490 Hz caused by the resonance in the second chamber. If you plot the pressure in the muffler at other dips, resonances in the other chambers appear.

Eigenmodes of a Room

Resonance can at times be a problem in everyday life. The low bass notes from the music system or home theater in the living room can shake the windows and make the floor vibrate. This happens only for certain frequencies—the eigenfrequencies of the room.

It is only in the low-frequency range that the eigenfrequencies are well separated. In the mid- and high-frequency ranges, the eigenfrequencies are packed so closely, with less than a halftone between them, that the individual resonances are insignificant for music and other natural sounds. Nevertheless, the music experience is affected by the acoustics of the room.

When designing a concert hall, it is extremely important to take the resonances into account. For a clear and neutral sound, the eigenfrequencies should be evenly spaced. For the home theater or music system owner, who cannot change the shape of the living room, another question is more relevant: Where should the speakers be located for the best sound?

Model Definition

For example, take a room with the dimensions 5 by 4 by 2.6 meters equipped with a TV set, two speakers, and a couch. To illustrate the effects on the music, compute all eigenfrequencies below 100 Hz together with the corresponding eigenmodes. The eigenmode shows the sound intensity pattern for its associated eigenfrequency. From the characteristics of the eigenmodes, you can draw some conclusions as to where the speakers should be placed.

DOMAIN EQUATIONS

Sound propagating in free air is described by the wave equation:

$$-\Delta p + \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0$$

where p is the pressure, and c is the speed of sound. If the air is brought into motion by a harmonically oscillating source, for example, a loudspeaker, only one frequency fexists in the room. For that reason it makes sense to look for a time-harmonic solution on the form

$$p = pe^{i\omega t}$$

The wave equation then simplifies to the Helmholtz equation for p, the amplitude of the acoustic disturbances:

$$\Delta \hat{p} + \frac{\omega^2}{c^2} \hat{p} = 0$$

BOUNDARY CONDITIONS

This model assumes that all boundaries—walls, floor, ceiling, and furniture —are perfectly rigid (sound hard boundaries).

ANALYTIC COMPARISON

It is possible to solve the simpler case of an empty room analytically. Each eigenfrequency corresponds to an integer triple (i, l, m):

$$f_{i,\,l,\,m}\,=\,\frac{c}{2}\sqrt{\left(\frac{i}{L_x}\right)^2+\left(\frac{l}{L_y}\right)^2+\left(\frac{m}{L_z}\right)^2}$$

The eigenmodes can be divided into three distinct classes:

- Eigenfrequencies with only one index different from zero give rise to axial modes, that is, plane standing waves between two opposite walls.
- If one index is zero, the mode is tangential.
- If all indices are different from zero, the mode is oblique.

Theoretical eigenvalues for a room without furniture are found in the following table.

MODE INDEX	$\omega^2/10^5$	MODE INDEX	$\omega^2/10^5$
0,0,0	0	0,1,1	2.444
1,0,0	0.465	2,1,0	2.584
0,1,0	0.726	0,2,0	2.902
1,1,0	1.191	1,1,1	2.908
0,0,1	1.718	I ,2,0	3.367
2,0,0	1.858	2,0,1	3.575
1,0,1	2.182	3,0,0	4.180

Results and Discussion

The relevant quantity when it comes to placing the loudspeakers is the amplitude of the standing pressure wave. A sound source excites an eigenmode the most if it is placed in one of the pressure antinodes for the mode. Conversely, with the source in a pressure node, the eigenmode remains silent.

All modes have local maxima in the corners of an empty room so speakers in the corners excite all eigenfrequencies. This simulation predicts eigenmodes that strongly resemble those of the corresponding empty room. The higher the frequency, the more the placing of the furniture matters. For instance, some of the high-frequency eigenmodes are located behind the couch.

In the strictest sense, the results of this simulation only apply to a room with perfectly rigid walls and nonabsorbing furniture. The prediction that speakers placed in the corners of the room excite many eigenmodes and give a fuller and more neutral sound, however, holds for real-life rooms.



Figure 2-5: The sound pressure distribution for $\omega = 569$ rad/s. The (negative of) the real part of the pressure is visualized as an isosurface plot, and the absolute value of the pressure as a boundary plot.

Model Library path: COMSOL_Multiphysics/Acoustics/eigenmodes_of_room

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Click Multiphysics.
- 2 Select 3D in the Space dimension list.
- **3** In the list of application modes, select **Acoustics** (in the **COMSOL Multiphysics** folder if your license includes additional modules), and in the **Acoustics** folder, select **Eigenfrequency analysis**.
- 4 Click Add.
- **5** Add geometries according to the following table by clicking the **Add Geometry** button.

NAME	SPACE DIMENSION	INDEPENDENT VARIABLES
Floor	2D	x y z
Table	2D	x y z
TV	2D	x y z

6 Click OK.

GEOMETRY MODELING

- I Click the **GeomI** tab.
- 2 Click the **Block** button to create a block with the following dimensions that you enter in the **Length** area:

PARAMETER	VALUE
Х	5
Y	4
Z	2.6

3 Click to deselect the **Highlight Face** toolbar button for a wireframe view.

Now create the cross section of the geometry on the floor of the room. The floor is by default an *x*-*y* plane at z = 0.
- I Go to the **Floor** work plane and click the **Zoom Extents** toolbar button.
- 2 From the **Options** menu, open the **Axes/Grid Settings** dialog box.
- **3** Click the **Grid** tab, clear the **Auto** check box, and change the grid according to the following table:

GRID	
x spacing	0.2
Extra x	1.7 2.1 4.5 4.7
y spacing	0.2
Extra y	1.7 2.3

- 4 Draw a rectangle with top left corner at (0.4, 3) and bottom right corner at (1.2, 1).
- **5** Draw eight squares with side 0.1 and upper left corners at (1.6, 1.7), (1.6, 2.4), (2.1, 1.7), (2.1, 2.4), (4.4, 1.7), (4.4, 2.4), (4.7, 1.7), and (4.7, 2.4).
- 6 From the Draw menu, choose Extrude.
- 7 In the Extrude dialog box, select all objects and set the Distance parameter to 0.4.
- 8 Click OK.



Figure 2-6: The geometry of the Floor work plane.

Now proceed to create the cross section of the geometry at the level of the table.

- I Go to the **Table** work plane.
- 2 Go to the Work-Plane Settings dialog box and change the z coordinate to 0.4. When done, click OK.
- 3 Click the Zoom Extents button on the Main toolbar.
- **4** In the **Axes/Grid Settings** dialog box, click the **Grid** tab and clear the **Auto** check box. Then change the grid spacing according to the following table:

GRID	
x spacing	0.2

y spacing 0.2

- **5** Draw a rectangle with top left corner at (1.6, 2.6) and bottom right corner at (2.2, 1.4).
- 6 Make sure the rectangle is selected and choose **Extrude** from the **Draw** menu. Set the distance to 0.1 and click **OK**.
- 7 Return to the **Table** work plane.
- 8 Draw a rectangle with top left corner at (0.4, 3) and bottom right corner at (1.2, 1), that is, following the blue projected contour.
- **9** Draw another rectangle with top left corner at (0.6, 2.8) and bottom right corner at (1.2, 1.2).
- **IO** Select the two last rectangles and click the **Difference** button.

II With the new composite object still selected, open the **Extrude** dialog box. Set **Distance** to 0.4 and **Displacement x** to -0.1 and click **OK**.



Figure 2-7: The geometry of the Table work plane.

Finally, create the geometry in the cross section of the TV.

- I Go the **TV** work plane.
- 2 Go to the Work-Plane Settings dialog box. Select the y-z plane and set the x coordinate to 4.4.
- 3 Click OK.
- **4** Click the **Projection of All 3D Geometries** button and after that the **Zoom Extents** button.
- **5** Use **Axes/Grid Settings** to change the grid as in the following table:

GRID	
x spacing	0.2
y spacing	0.2

6 Draw the TV set as a rectangle with top left corner at (1.6, 1.0) and bottom right corner at (2.4, 0.4).

- **7** Draw the left speaker as a rectangle with top left corner at (0.8, 1.0) and bottom right corner at (1.2, 0.0).
- 8 Draw the right speaker with the same size, but with top left corner at (2.8, 1.0).
- 9 Select all objects and open the Extrude dialog box. Set distance to 0.4 and click OK.

IO In the 3D geometry, select all objects and click the **Difference** button.

Now the room is completed and should look like in Figure 2-8.



Figure 2-8: The 3D geometry.

PHYSICS SETTINGS

Boundary Conditions

Use the Sound hard boundary (wall) boundary condition for all boundaries.

Subdomain Settings Use the default subdomain settings.

MESH GENERATION

Click the **Initialize Mesh** button to initialize the mesh.

COMPUTING THE SOLUTION

- I Make sure that **Eigenfrequency** is selected from the **Solver** list in the **Solver Parameters** dialog box.
- 2 To get 6 eigenfrequencies around 90 Hz enter 6 in the Desired number of eigenfrequencies edit field, this is default, and 90 in the Search for eigenfrequencies around edit field.
- 3 Click OK to close the Solver Parameters dialog box.
- 4 Click the **Solve** button to compute the solution.

POSTPROCESSING AND VISUALIZATION

The default plot style is a slice plot showing the sound pressure in five equally spaced slices. Try looking at some of the eigenmodes. The first eigenvalue in the list is very small. Its true value is zero, corresponding to the solution without any sound. The next few are axial and tangential modes. To get a better view of the more complicated eigenmodes, you can do a combined surface and boundary plot (see Figure 2-5).

- I In the **Suppress Boundaries** dialog box in the **Options** menu, suppress the boundaries 1, 2, and 4.
- 2 On the General page of the Plot Parameters dialog box, clear the Slice check box and select the Isosurface, Boundary, and Geometry edges check boxes in the Plot type area. In the Eigenfrequency list, select the eigenvalue of about 90.6.
- 3 Click the **Boundary** tab and type abs(p) in the **Expression** edit field.
- 4 Click the Isosurface tab and type -p in the Expression edit field.
- 5 Click the Scene Light and Perspective Projection buttons in the Camera toolbar.
- 6 Click OK.

This particular mode is concentrated behind the couch. Try moving around the room and looking at other eigenmodes. It is possible that you can identify some of them with the exact solutions for the case of an empty room.

Chemical Engineering Models

This section contains a model of a tubular reactor and a model of surface diffusion and surface reactions coupled to transport of species to the reacting surface—a comprehensive set of chemical engineering models covering mass balances, energy balances, and momentum balances is available in the *Model Library* that comes with the Chemical Engineering Module.

Transport and Adsorption

This model demonstrates one of the unique features of COMSOL Multiphysics: the possibility to model phenomena defined in different numbers of dimensions in a fully coupled manner.

Whereas in most cases you define the reaction rate expression as a function of the concentrations of the reactants and products, in adsorption reactions it is also necessary to model the surface concentrations of the active sites or surface complex. This implies that the mass balance in the bulk of the reactor must be coupled to the mass balance for species present only at the surface of the device. This device could be a catalyst, a biochip, a semiconductor component, or any process with surface-specific species.

In this particular case, the model involves a small parallel-plate reactor with an active surface. It is a simple example of surface diffusion and surface reactions coupled to the transport of species between the reacting surface and the surrounding volume. Processes of this kind are found, for instance, in heterogeneous reactors and biacore chips.

Model Definition

The geometry of the domain appears in Figure 3-1.



Figure 3-1: The modeled domain is a parallel plate reactor with an active surface where you want to model the concentration of surface species.

The first approximation you can make is to reduce the 3D geometry to a 2D approximation, which is reasonable if the variations in concentration are small along the depth of the domain.

DOMAIN EQUATIONS

The reaction at the active surface is given as

$$\begin{array}{c} k_{\mathrm{ads}} \\ c + \theta \Leftrightarrow c_{\mathrm{s}} \\ k_{\mathrm{des}} \end{array}$$

where

- *c* is the bulk concentration (mol/m^3)
- θ is the surface concentration of active sites (mol/m²)
- c_s is the surface concentration of adsorbed species (mol/m²)

- k_{ads} is the rate constant for the forward reaction (m³/(mol·s))
- k_{des} is the rate constant for the backward reaction (1/s)

Note: The surface concentration is in moles per unit surface.

The material balance for the surface, including surface diffusion and the reaction rate expression for the formation of the adsorbed species, c_s , is:

$$\frac{\partial c_{\rm s}}{\partial t} + \nabla \cdot (-D_{\rm s} \nabla c_{\rm s}) = k_{\rm ads} c \theta - k_{\rm des} c_{\rm s}$$

where D_s represents surface diffusivity. However, the concentration of active sites is equal to the difference between the total concentration of active sites and the number of sites occupied by the adsorbed species. This gives the following equation for the reaction rate:

$$\frac{\partial c_{s}}{\partial t} + \nabla \cdot (-D_{s} \nabla c_{s}) = k_{ads} c(\theta_{0} - c_{s}) - k_{des} c_{s}$$
(3-1)

In the above equation, θ_0 represents the total number of active sites available on the surface of the catalyst. Equation 3-1 also defines the units of the rate constants: k_{ads} and k_{des} . The initial condition is that the concentration of adsorbed species is zero at the beginning of the process:

$$c_{\rm s} = 0$$

The equation for the surface-reaction expression includes the concentration of the bulk species, c, at the position of the catalyst surface. Thus you must solve the equation for the surface reaction in combination with the mass balance in the bulk. The coupling between the mass balance in the bulk and the surface is obtained as a boundary condition in the bulk's mass balance. This condition sets the flux of c at the boundary equal to the rate of the surface reaction and is presented below. The transport in the bulk of the reactor is described by a convection-diffusion equation:

$$\frac{\partial c}{\partial t} + \nabla \cdot (-D\nabla c + c\mathbf{u}) = 0$$
(3-2)

The initial condition sets the concentration in the bulk at t = 0:

 $c = c_0$

In the above equation, D denotes the diffusivity of the reacting species, c is its concentration, and **u** is the velocity. In this case, the velocity in the x direction equals 0 while the velocity in the y direction comes from the analytical expression for fully developed laminar flow between two parallel plates:

u =
$$(0, v_{\max} \left[1 - \left(\frac{(x - 0.5\delta)}{0.5\delta} \right)^2 \right])$$

Here, δ is the distance between the plates, and v_{max} is the maximum local velocity. We assume that the origin for *x* is at the left edge of the model depicted in Figure 3-1.

BOUNDARY CONDITIONS

The boundary conditions for the material balance for the surface species are insulating conditions according to:

$$\mathbf{n} \cdot (-D_{\mathbf{s}} \nabla c_{\mathbf{s}}) = 0 \tag{3-3}$$

For the bulk, the boundary condition at the active surface couples the rate of the reaction at the surface with the flux of the reacting species and the concentration of the adsorbed species and bulk species:

$$\mathbf{n} \cdot (-D\nabla c + c\mathbf{u}) = -k_{ads}c(\theta_0 - c_s) + k_{des}c_s$$

The other boundary conditions for the bulk problem are:

- Inlet: $c = c_0$
- Outlet: $\mathbf{n} \cdot (-D\nabla c + c\mathbf{u}) = \mathbf{n} \cdot c\mathbf{u}$
- Insulation: $\mathbf{n} \cdot (-D\nabla c + c\mathbf{u}) = 0$

Modeling in COMSOL Multiphysics

This model deals with a phenomenon occurring in a 2D domain (convectiondiffusion) coupled to another phenomenon occurring only at the 1D boundary of the 2D domain (diffusion-reaction). The COMSOL Multiphysics implementation is straightforward in every step except for the definition of the 1D problem, which requires some mathematical background. You need to use weak-form application modes for modeling PDEs on boundaries, edges, or points. To add the 1D boundary equation in the 2D model, use the Weak Form, Boundary application mode. This enables you to formulate the problem with the weak form instead of with the PDE form. An examination of the spatial diffusion on the boundary shows that only the flux tangential to the surface is defined. Therefore, in the weak-form specification you must make use of the special *tangential derivative variable*. For more information about how to specify equations on the weak form, refer to the chapter "Deriving the Weak Form" on page 310 in the *COMSOL Multiphysics Modeling Guide*.

You define the two equations using the multiphysics feature of COMSOL Multiphysics. Select the two application modes, corresponding to the 2D and 1D material balances, in the Model Navigator.



Results

Figure 3-2: The concentration of the reacting species, c, after 2 seconds of operation in the 2D cross section (top) and along the active surface (bottom).

The upper plot in Figure 3-2 shows the concentration, c, of the reacting species in the 2D subdomain after 2 seconds of operation. The reaction is very fast and almost

reaches steady state in that time frame. The lower plot, displaying the concentration along a vertical cross section of the active surface, shows that the concentration distribution exhibits edge effects at both ends of the catalyst. The higher concentration near y = 0 is easy to explain because this is the position closest to the inlet, and this end is therefore continuously supplied with fresh reactant. The increase in concentration at the end closer to the outlet is due to radial diffusion; the edge of the surface can receive diffusion in all directions within a 90° angle without having to "compete" for the reactant supply with other parts of the catalysts. This effect also appears at the edge close to the inlet.

The concentration of adsorbed species, c_s , shows a similar spatial distribution. However, while the concentration of the reactant decreases with time, the concentration of adsorbed species increases. You can also see that the slight spatial diffusion evens out the concentration gradients somewhat (see Figure 3-3).



Figure 3-3: The concentration of adsorbed species increases with time. The edge effect appears at both edges due to the increased supply of reactants. This figure displays the concentration after 0.05, 0.5, 1.0, 1.5, and 2.0 s.

The concentration of reacting species decreases while the surface concentration of the adsorbed species increases with time. This implies that the surface reaction rate decreases with time. You can see this effect in Figure 3-4, which also shows that after

0.5 s the reaction rate almost reaches steady-state. The upper curve shows the reaction rate after 0.05 s, while the lower curve represents the curves for 0.5, 1.0, 1.5, and 2.0 s, which are all on top of each other.



Figure 3-4: Surface reaction rate at the active surface. The largest reaction rate is obtained initially and is at the edges of the active surface. The reaction process almost reaches steady state at 0.5 s.

Model Library path: COMSOL_Multiphysics/Chemical_Engineering/ adsorption

Modeling Using the Graphical User Interface

The following pages describe how to solve this problem using the graphical user interface.

MODEL NAVIGATOR

I Start COMSOL Multiphysics.

- 2 In the Model Navigator, click the Multiphysics button and select 2D in the Space dimension list.
- 3 Highlight the application mode COMSOL Multiphysics>Convection and Diffusion>Convection and Diffusion>Transient analysis. Click the Add button.
- 4 Select the application mode COMSOL Multiphysics>PDE Modes>Weak Form, Boundary. Type cs in the Dependent variables edit field. Click the Add button.
- 5 Click OK.

Model Navigator	E Contraction of the second se
New Model Library User Models Open Settings	
Space dimension: ZD Structural Mechanics Plane Strain Static analysis Static analysis Figenfrequency analysis Classical PDEs Coefficient Form Structural Mechanics E PDE Modes Structural Mechanics E Structural Mechanics Structura	Multiphysics Add Remove Geom1 (2D) Corvection and Diffusion (cd) Wesk Form, Boundary (wb)
	Dependent variables: cs Application Mode Properties Add Geometry Add Frame
Dependent variables: u Application mode name: wb2 Element: Lagrange - Quadratic +	Ruling application mode: Convection and Diffusion (cd) Multiphysics
	OK Cancel Help

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- 2 Define the following constants in the **Constants** dialog box:

NAME	EXPRESSION	DESCRIPTION
c0	1000[mol/m^3]	Initial concentration
kads	1e-6[m^3/(mol*s)]	Forward rate constant
kdes	1e-9[1/s]	Backward rate constant
theta0	1000[mol/m^2]	Active site concentration
Ds	1e-11[m^2/s]	Surface diffusivity
D	1e-9[m^2/s]	Gas diffusivity
v_max	1[mm/s]	Maximum velocity
delta	0.1[mm]	Channel width

- 3 Click OK.
- **4** Open the **Axes/Grid Settings** dialog box from the **Options** menu. Specify the following settings and click **OK**.

POSITION	X-Y LIMITS
x min	-15e-5
x max	25e-5
y min	-15e-5
y max	25e-5

GEOMETRY MODELING

I Hold down the shift key and click the **Rectangle/Square** button on the Draw toolbar. Enter the values listed below in the corresponding edit fields and click **OK**.

EDIT FIELD	VALUE
Width	1e-4
Height	3e-4
x	0
у	-1e-4

- 2 Click the Zoom Extents button on the Main toolbar.
- **3** Click the **Point** button and click on the coordinate (1e-4, 0) in the drawing area. Click the **Point** button again and click on (1e-4, 1e-4).

PHYSICS SETTINGS

Expression Variables

- I On the **Options** menu, point to **Expressions** and then click **Boundary Expressions**.
- 2 Select Boundary 5 in the Boundary selection list and enter the following expression.

NAME	EXPRESSION
react_surf	kads*c*(theta0-cs)-kdes*cs

3 Click OK.

4 On the Options menu, select Expressions and then click Subdomain Expressions.

5 Select Subdomain 1 in the **Subdomain selection** list and enter the following expression.

NAME	EXPRESSION
v_lam	v_max*(1-((x-0.5*delta)/(0.5*delta))^2)

6 Click OK.

Boundary Conditions

- I From the Multiphysics menu, select I Convection and Diffusion (cd).
- 2 From the Physics menu, choose Boundary Settings.

3 Enter boundary conditions according to the following table:

SETTINGS	BOUNDARIES 1, 4, 6	BOUNDARY 2	BOUNDARY 3	BOUNDARY 5
Туре	Insulation/ Symmetry	Concentration	Convective flux	Flux
c ₀		c0		
N ₀				-react_surf

4 Click OK.

Subdomain Settings

- I From the Physics menu, choose Subdomain Settings.
- I Select Subdomain 1 in the Subdomain Settings dialog box.
- 2 Specify the settings as in the following table:

QUANTITY	VALUE/EXPRESSION
δ_{ts}	1
D isotropic	D
R	0
u	0
v	v_lam

3 Click the **Init** tab and type c0 in the **Concentration**, c edit field (c(t₀)).

4 Click OK.

Boundary Conditions

Perform this step to specify the transport equation stated in Equation . Do so in the **Boundary Settings** dialog box because the catalytic surface is a boundary of the 2D geometry.

- I In the Multiphysics menu, select 2 Weak Form, Boundary (wb).
- 2 From the Physics menu, choose Boundary Settings.
- **3** Select the **Select by group** check box, then click in the **Boundary selection** list to select all boundaries (1–6). Clear the **Active in this domain** check box.
- 4 Clear the Select by group check box, then select Boundary 5. Check the Active in this domain check box to enable the weak form application mode only on this boundary.
- 5 Click the **Init** tab and then type 0 in the cs(t₀) edit field.
- 6 Click the Weak tab. Make the following specifications:

TERM	VALUE/EXPRESSION
weak	Ds*(-test(csTx)*csTx-test(csTy)*csTy)+ test(cs)*(react_surf-cst)
dweak	0
constr	0

7 Click OK.

By adding T to all the weak variables you get the tangential components discussed in the section "Modeling in COMSOL Multiphysics" on page 33. The lowercase t indicates the derivative with respect to time.

Point Settings

To enforce the boundary condition in Equation on the surface equation you must open the **Point Settings** dialog box from the **Physics** menu and select Points 4 and 5. In this case the default settings are correct, that is, the **weak**, **dweak**, and **constr** edit fields should all contain 0, so just leave them as is.

MESH GENERATION

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 In the Free Mesh Parameters dialog box, click the Boundary tab and select Boundary 5 from the list.
- **3** Type **1.5e-6** in the **Maximum element size** edit field.
- 4 Click Remesh, then click OK.

COMPUTING THE SOLUTION

- I Click the Solver Parameters button.
- 2 Select Time dependent in the Solver list.
- **3** In the **Time stepping** area, type 0:0.05:2 in the **Times** edit field.

- 4 Click OK.
- 5 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To generate the right plot in Figure 3-2:

- I From the **Postprocessing** menu, choose **Domain Plot Parameters**.
- **2** Go to the **General** tab and click the **Line/Extrusion plot** button. Select **2** in the **Solutions to use** list. Customize the plot with text labels by clicking the **Title/Axis** button.
- 3 Click the Line/Extrusion tab and then select Convection and Diffusion (cd)> Concentration, c from the Predefined quantities list.
- **4** From the **Boundary selection** list select Boundary 5.
- 5 Under x-axis data, click the upper option button. Select y from the adjacent list.
- 6 Click OK.

To generate Figure 3-3, repeat the previous operations but with these changes:

- I On the **General** page, go to the **Solutions to use** list and hold down the Ctrl key to select the times 0.05, 0.5, 1.0, 1.5, and 2.0.
- 2 Click the Line/Extrusion tab and then select Weak Form, Boundary (wb)>cs from the Predefined quantities list.

To generate Figure 3-4, repeat the previous operations but with this change:

On the Line/Extrusion tab, type react_surf in the Expression edit field.

Tubular Reactor

Introduction

This example studies an elementary, exothermic, 2nd-order reversible reaction

$$A + B \Leftrightarrow 2C \tag{3-4}$$

in a tubular reactor (liquid phase, laminar flow regime). The reactor is equipped with a cooling jacket to limit the temperature increase due to the exothermic nature of the reaction and avoid an explosion. The model is described by the material balances for the species involved and the energy balances for the reactor and the cooling jacket.

Additional background to this model is available in Section 8.9 of the Fourth Edition of *Elements of Chemical Reaction Engineering* by H.S. Fogler.

Model Definition

Figure 3-5 illustrates the reactor geometry. By assuming that the variations in the angular direction around the centerline are negligible, you can use a 2D axisymmetric model.



Figure 3-5: Geometry for the 2-dimensional rotationally symmetric models.

The model geometry consists of an inlet boundary, an outlet boundary, a reactor wall facing the cooling jacket, and the symmetry axis r = 0. The reactor's radius is R = 8 cm its length L = 1 m.

Assuming the diffusivities for the three species to be identical, you can model the reactor using three differential equations:

- A material-balance partial differential equation (PDE) for one of the species. As noted in the next section, separate material-balance equations are not necessary for the other two species if the diffusivities are identical.
- An energy-balance PDE for the reactor core.
- An energy-balance ordinary differential equation (ODE) for the cooling jacket. Under the assumptions that the coolant flow is turbulent and that turbulent mixing eliminates any temperature differences in the radial direction, only axial temperature variations are present in the cooling jacket. For this reason, the cooling jacket is not included in the model geometry.

The material and energy balances in the reactor are automatically set up in COMSOL Multiphysics by selecting the proper application modes. The ordinary differential equation that describes the energy balance in the cooling jacket is manually defined as a boundary equation.

The material balance and energy balances for the reactor are described by the equations

$$\nabla \cdot (-D\nabla c_{A} + \mathbf{u}c_{A}) = r_{A}$$

$$\nabla \cdot (-k\nabla T + \mathbf{u}\rho C_{p}T) = \Delta H_{Rx}r_{A}$$
(3-5)

In the material-balance equation for $c_A \text{ (mol/m}^3)$, the concentration of species A, $D \text{ (m}^2/\text{s})$ denotes the diffusion coefficient, **u** (m/s) the flow velocity, and $r_A \text{ (mol/} (\text{m}^3 \cdot \text{s}))$ the reaction rate. In the energy-balance equation for the reactor temperature, T (K), the additional parameters are the thermal conductivity, $k \text{ (W/(m} \cdot \text{K))}$, the fluid density, $\rho \text{ (kg/m}^3)$, the specific heat capacity, $C_p \text{ (J/(kg} \cdot \text{K))}$, and the reaction enthalpy $\Delta H_{Rx} \text{ (J/mol)}$.

The boundary conditions for Equations 3-5 are as follows (see Figure 3-5 for the appropriate references):

$$\begin{split} c_A(r,0) &= c_{A0} \quad \frac{\partial c_A}{\partial r}(R,z) = \frac{\partial c_A}{\partial r}(0,z) = 0 \quad \frac{\partial c_A}{\partial z}(r,L) = 0 \\ T(r,0) &= T_0 \quad \frac{\partial T}{\partial r}(R,z) = \frac{\partial T}{\partial r}(0,z) = 0 \quad \frac{\partial T}{\partial z}(r,L) = 0 \end{split}$$
(3-6)

Here c_{A0} and T_0 denote the initial concentration and temperature, respectively. For the flow velocity, assume a laminar profile:

$$u = u_{\max} \left(1 - \left(\frac{r}{R}\right)^2 \right) \tag{3-7}$$

Here $u_{\text{max}} = 2u_0$, where the average velocity, u_0 (m/s), is calculated as the ratio between the total volume flow rate, v_0 (m³/s), and the reactor's cross-sectional area, πR^2 .

In this model, assume that the species A, B, and C have the same diffusivity. This implies that you only need to solve one material balance, because you can find the other species' concentrations through stoichiometry. Specifically, assuming the initial concentration for species C to be zero, the concentrations are given by

$$c_{\rm A} = (1 - x_{\rm A})c_{\rm A0}$$
 $c_{B} = c_{B0} - x_{\rm A}c_{\rm A0}$ $c_{\rm C} = 2x_{\rm A}c_{\rm A0}$ (3-8)

where x_A is the conversion of species A.

In the COMSOL Multiphysics Chemical Engineering Module, you could have modeled all three species using their true multicomponent diffusivities. These assumptions give the reaction-rate expression

$$r_{\rm A} = -A \exp\left(\frac{E}{R_{\rm g}T}\right) \left(c_{\rm A}c_{\rm B} - \frac{c_{\rm C}^2}{K_{\rm eq}}\right)$$
(3-9)

where $A (m^3/(\text{mol}\cdot s))$ refers to the rate constant, E (J/mol) is the activation energy, $R_g = 8.314 \text{ J}/(\text{mol}\cdot\text{K})$ denotes the ideal gas constant, T (K) is the temperature, and the equilibrium constant is

$$K_{\rm eq} = K_{\rm eq0} \exp\left(\Delta H_{Rx} R_{\rm g} \left(\frac{1}{303 \text{ K}} - \frac{1}{T}\right)\right) \tag{3-10}$$

The energy-balance ODE for the coolant reads

$$\frac{\partial T_{\rm c}}{\partial z} = \frac{2\pi R U_k (T - T_{\rm c})}{C_{pc} m_{\rm c}}$$
(3-11)

where T_c is the coolant temperature (K), U_k (W/(m²·K)) is the total heat-transfer coefficient between the reactor and the cooling jacket, rate, C_{pc} (J/(kg·K)) is the coolant's specific heat capacity, and m_c (kg/s) represents its mass flow. You can neglect the heat conduction in the cooling jacket and thus assume that heat transport takes place only through convection.

As a boundary condition, set the temperature of the incoming cooling fluid:

$$T_{\rm c}(0) = T_{\rm c0} . \tag{3-12}$$

Table 3-1 summarizes the relevant input data.

PROPERTY	VALUE
D	10 ⁻⁹ m ² /s
k	0.559 W/(m·K)
ρ	1000 kg/m ³
C_p	4180 J/(kg·K)
ΔH_{Rx}	-83,680 J/mol
T_0	313 K
c_{A0}	500 mol/m ³
$c_{\rm B0}$	500 mol/m ³
v_0	5·10 ⁻⁴ m ³ /s
Α	1.65·10 ¹¹ m ³ /(mol·s)
Ε	95,238 J/mol
$K_{ m eq0}$	10 ³
U_k	1300 W/(m ² ·K)
C_{pc}	4180 J/(kg·K)
m _c	0.01 kg/s
$T_{\rm c0}$	298 K

TABLE 3-1: MODEL INPUT DATA

Results

From the plots in Figure 3-6 you can see how the conversion and temperature vary inside the reactor. Due to a larger residence time, you find the highest conversion close to the wall (right side). Also, the cooling effect of the wall is evident from the temperature plot. Nevertheless, the temperature maximum is located close to the wall because the temperature depends strongly on the relation between the convected heat and the heat produced in the reaction. Due to the laminar-flow velocity profile, the

fluid close to the reactor wall but not in contact with the cooling jacket is heated during a longer time by the exothermic reactions.



Figure 3-6: Conversion (left) and temperature (right) in the reactor.

Figure 3-7 shows how the coolant temperature increases along the length of the reactor. The temperature increases rapidly close to the inlet, but as the reaction rate decreases due to the depletion of reactants so does the rate of temperature increase.



Figure 3-7: Coolant temperature along the length of the reactor.

Model Library path:

COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor

MODEL NAVIGATOR

- I Go to the Model Navigator and select Axial symmetry (2D) from the Space dimension list.
- 2 Click the Multiphysics button in the lower-right corner.
- **3** Open the **COMSOL Multiphysics** folder and select the **Convection and Diffusion** application mode from the **Convection and Diffusion** folder.
- 4 In the Dependent variables edit field, type cA. Click Add.
- 5 Add the Convection and Conduction application mode from the Heat Transfer folder.
- 6 From the Space dimension list, select 2D.
- 7 Open the COMSOL Multiphysics>PDE Modes folder and select Weak Form, Boundary.
- 8 Change the name of the dependent variable to Tc, then click Add.
- 9 Click OK.

OPTIONS AND SETTINGS

- I Go to the **Options** menu and choose **Constants**.
- 2 Make the following entries in the Constants dialog box and then click OK.

NAME	EXPRESSION	DESCRIPTION
D	1e-9[m^2/s]	Diffusivity
k	0.559[W/(m*K)]	Thermal conductivity, reactor
rho	1000[kg/m^3]	Fluid density, reactor
Ср	4180[J/(kg*K)]	Specific heat capacity, reactor
dHRx	-83680[J/mol]	Reaction enthalpy
то	320[K]	Initial temperature, reactor
cA0	500[mol/m^3]	Initial concentration, species A
cB0	500[mol/m^3]	Initial concentration, species B
R	8[cm]	Reactor radius
v0	0.0005[m^3/s]	Total flow rate, reactor
u0	v0/(pi*R^2)	Mean axial flow velocity
E	95238[J/mol]	Activation energy
A	1.65e11[m^3/(mol*s)]	Rate constant frequency factor
Rg	8.314[J/(mol*K)]	Ideal gas constant
Keq0	1000	Equilibrium constant at 303 K

NAME	EXPRESSION	DESCRIPTION
Uk	1300[W/(m^2*K)]	Heat transfer coefficient
Tc0	298[K]	Coolant temperature, inlet
Срс	4180[J/(kg*K)]	Specific heat capacity, coolant
mc	0.01[kg/s]	Total mass-flow rate, coolant

3 Go to the Options menu and choose Expressions>Scalar Expressions.

4 Make the following entries in the Scalar Expressions dialog box; when done, click OK.

NAME	EXPRESSION	DESCRIPTION	
uz	2*u0*(1-(r/R)^2)	Axial flow velocity	
хA	(cA0-cA)/cA0	Conversion, species A	
сВ	cB0-cA0*xA	Concentration, species B	
cC	2*cA0*xA	Concentration, species C	
rA	-A*exp(-E/(Rg*T)) *(cA*cB-cC^2/Keq)	Reaction rate	
Keq	Keq0*exp(dHRx/Rg *(1/303[K]-1/T))	Equilibrium constant	
Q	(-rA)*(-dHRx)	Heat production	

GEOMETRY MODELING

- I Go to the Draw menu and select Specify Objects>Rectangle.
- 2 Type 0.08 in the Width edit field and 1 in the Height edit field. Click OK.
- 3 Click Zoom Extents to center the geometry.

PHYSICS SETTINGS

Subdomain Settings—Convection and Diffusion

- I From the Multiphysics menu, select Convection and Diffusion (cd).
- 2 From the Physics menu, select Subdomain Settings.
- **3** In the **Subdomain Settings** dialog box, select Subdomain 1 and type D in the **Diffusion coefficient** edit field.
- 4 In the Reaction rate edit field, type rA.
- **5** In the **z-velocity** edit field, type uz.
- 6 Still in the Subdomain Settings dialog box, click the Init tab.
- 7 In the edit field for $cA(t_0)$, enter the initial value cA0.
- 8 Click OK.

Boundary Conditions—Convection and Diffusion

I From the Physics menu, choose Boundary Settings.

2 In the Boundary Conditions dialog box enter the following boundary conditions.

SETTINGS	BOUNDARIES I, 4	BOUNDARY 2	BOUNDARY 3
Туре	Insulation/Symmetry	Concentration	Convective flux
cA ₀	-	cA0	-

3 Click OK.

Subdomain Settings—Convection and Conduction

- I From the Multiphysics menu, choose Convection and Conduction (cc).
- 2 From the Physics menu, choose Subdomain Settings and select Subdomain 1.
- 3 In the Thermal conductivity edit field, type k.
- 4 In the **Density** edit field, type rho.
- 5 In the Heat capacity edit field, type Cp.
- 6 In the Heat source edit field, type Q.
- 7 In the **z-velocity** edit field, type uz.
- 8 Still in the Subdomain Settings dialog box, click the Init tab.
- 9 In the edit field for **T**(t₀), enter the initial value T0.

IO Click OK.

Boundary Conditions—Convection and Conduction

I From the Physics menu, choose Boundary Settings.

2 In the Boundary Conditions dialog box enter the following boundary conditions.

SETTINGS	BOUNDARY I	BOUNDARY 2	BOUNDARY 3	BOUNDARY 4
Туре	Axial symmetry	Heat flux	Convective flux	Heat flux
90	-	rho*Cp*T0*uz	-	-Uk*(T-Tc)

Because Tc is the dependent variable of a PDE application mode, it does not have an associated unit. Therefore, if the **Highlight unexpected units** check box is selected in your modeling session, the **Unit** label for the **q**₀ edit field appears in red. Provided you specify the boundary condition for Tc using the chosen base unit system (specified under **Physics>Model Settings**; the default choice is **SI**), you can safely ignore this warning.

3 Click OK.

Boundary Conditions-Weak Form, Boundary

- I From the Multiphysics menu, choose Weak Form, Boundary (wb).
- 2 From the Physics menu, choose Boundary Settings.
- **3** Select Boundaries 1–3 in the **Boundary selection** list and clear the **Active in this domain** check box.
- **4** Select Boundary 4 and click the **Weak** tab.
- 5 In the weak edit field, type Tc_test*(TcTz-2*pi*R*Uk*(T-Tc)/(Cpc*mc)).
- 6 Click OK.
- 7 From the Physics menu, choose Point Settings.
- 8 Select Point 3 from the Point selection list.
- **9** Set the inlet temperature of the cooling water by typing Tc-Tc0 in the **constr** edit field.
- IO Click OK.

MESH GENERATION

- I From the Mesh menu, choose Mapped Mesh Parameters.
- **2** Click the **Boundary** tab.
- **3** Select Boundaries 1 and 4 from the **Boundary selection** list by left-clicking and holding down the Ctrl-key.
- 4 Select the Constrained edge element distribution check box, then select Edge vertex distribution. In the corresponding edit field, type [logspace(-2,1,100)./10-1e-3 1].
- 5 Select Boundaries 2 and 3 in the Boundary selection list.
- 6 Select the Constrained edge element distribution check box then select Edge vertex distribution. In the corresponding edit field, type 1+(log(linspace(1/61,1-1/61,61))./ max(-log(linspace(1/61,1-1/61,61)))).
- 7 Click Remesh, then click OK.

The generated mesh uses a dense mesh close to the cooling wall to resolve the large gradients in temperature, and also close to the reactor inlet to resolve the large gradients in composition.

COMPUTING THE SOLUTION

I Click the Solver Parameters button on the Main toolbar.

- 2 Make sure the Stationary solver is selected.
- 3 Click OK.
- 4 Click the Solve button on the Main toolbar to start the simulation.

POSTPROCESSING AND VISUALIZATION

The default plot displays the concentration profile.

To see the resulting conversion or temperature profiles, follow these steps:

- I Click the Plot Parameters button on the Main toolbar.
- **2** To see the conversion or temperature profile, go to the **Surface** page and type xA (for the conversion) or T (for the temperature) in the **Expression** edit field.
- 3 Click OK to close the Plot Parameters dialog box.
- **4** Use the **Zoom Window** tool on the Main toolbar to get a closer look at the results (see Figure 3-6).

To plot the coolant temperature throughout the length of the reactor (see Figure 3-7), follow these steps:

- I From the **Postprocessing** menu, choose **Domain Plot Parameters**.
- 2 Click the Line/Extrusion tab and select Boundary 4 from the Boundary selection list.
- 3 From the Predefined quantities list, select Weak Form, Boundary (wb)>Tc.
- 4 From the x-axis data list, select z.
- 5 Click OK.

To investigate the average outlet conversion, concentration, and temperature follow the steps below:

- I From the Postprocessing menu, choose Boundary Integration.
- 2 Select Boundary 3 in the Boundary selection list.
- **3** Select the **Compute surface integral (for axisymmetric modes)** check box.
- 4 Type 1 in the Expression edit field to get the cross-sectional area of the reactor.
- 5 Click Apply. The value of the integral (roughly 0.0201 m²) appears in the message log at the bottom of the user interface.
- **6** Type xA in the **Expression** edit field, then click **OK**. To get the average outlet conversion, divide the result with the reactor's cross-sectional area.
- 7 Follow the same procedure using the expressions cA and T to find the average outlet concentration and temperature, respectively.

Electromagnetics Models

This chapter contains a few electromagnetics models—more comprehensive sets are available in the model libraries that comes with the AC/DC Module and RF Module. For introductory electromagnetics models and information about the application modes for electromagnetics, see "Electromagnetics" on page 75 in the *COMSOL Multiphysics Modeling Guide*.

Several physical quantities are fundamental to electromagnetics. They are:

- Electric field intensity, E
- Electric flux density or electric displacement, D
- Magnetic flux density, B
- Magnetic field intensity, H
- Current density, J
- Electric charge density, p

The governing partial differential equations for all macroscopic electromagnetic problems are well known—they are the four *Maxwell's equations*. Of these relationships, the first three are *Faraday's law*, *Ampère's circuital law*, and *Gauss' Law*. The fourth equation states that isolated magnetic charges (monopoles) do not exist in nature.

In electromagnetics, *constitutive relations* specify material-dependent interactions between fundamental physical quantities such as **E** and **D**. The definition of boundary conditions, taken together with Maxwell's equations and the constitutive relations, make it possible to describe any macroscopic electromagnetic problem. For an introduction to the theory of electromagnetics, see the section "Fundamentals of Electromagnetics" on page 77 in the *COMSOL Multiphysics Modeling Guide*.

Electrostatic Potential Between Cylinders

This example model demonstrates each step in building and solving a 3D electrostatics model. The analysis shows the electric potential between two cylinders in a vacuum.

Model Definition

The first step is to define the overall model and draw its geometry. This model consists of two cylinders with opposite electrostatic potentials. The potential difference between them induces an electric field in the vacuum. The electrostatic scalar potential *V* is related to an electric field as $\mathbf{E} = -\nabla V$. Gauss' law, $\nabla \cdot (\epsilon \mathbf{E}) = \rho$, gives $-\nabla \cdot (\epsilon \nabla V) = \rho$.

This equation includes a material property and a source term:

- ε is the permittivity.
- ρ is the space charge density.

To simulate the vacuum chamber, create a box in which there is no atmosphere or gas of any kind. In addition, assume that the potential is zero on the bottom boundary and that there is no electric field normal to the other exterior boundaries.

Results

The plot in Figure 4-1 on page 56 combines a surface plot of the potential with a streamline plot of the electric field. The surface plot shows the electric potential on the cylinder and box faces. As expected, the electric field lines run from cylinder to cylinder and from the cylinders to the grounded bottom plate. The electric field does not penetrate the top and side walls.



Figure 4-1: The electric potential (surface) and the electric field (streamlines).

Modeling Using the Graphical User Interface

Start COMSOL Multiphysics using the procedure for the operating system on your computer. On Windows double-click the **COMSOL Multiphysics** icon to start COMSOL Multiphysics and open the **Model Navigator**.

Model Library path: COMSOL_Multiphysics/Electromagnetics/ potential_between_cylinders

MODEL NAVIGATOR

For this model use the Electrostatics application mode.

- I In the Model Navigator click on the New tab.
- 2 Select 3D in the Space dimension list.

- **3** In the list of application modes, open the **COMSOL Multiphysics** folder and then the **Electromagnetics** folder. Select **Electrostatics** from the list of electromagnetics application modes.
- 4 Accept the default element type (quadratic Lagrange elements), which is suitable for most cases.
- 5 Click OK.



The Electrostatics application mode provides predefined equations and boundary conditions that make it easy to create this model, but you must still attend to a few details.

OPTIONS AND SETTINGS

In this model the permittivity of free space, ε_0 , defines the permittivity in the domain. It is not necessary to enter this variable because it is a predefined application scalar variable in the Electrostatics application mode. If you prefer to use another value rather than the standard SI unit value, choose **Scalar Variables** from the **Physics** menu. Doing so, however, changes the underlying formulation for electrostatics, so we recommend that you use the predefined value.

GEOMETRY MODELING

Next define the model geometry. For this example, create a box measuring $2 \times 2 \times 2$ meters, and then place two cylinders inside that box. When you exit the **Model**

Navigator to begin setting up a model, COMSOL Multiphysics enters Draw mode with access to the drawing tools.

To define the model geometry, begin with the box:

- I Open the **Block** dialog box by clicking on the appropriate symbol on the Draw toolbar.
- **2** In that dialog box, go to the **Base** area and click the **Center** button to use centered base coordinates.
- **3** In the **Length** area, type **2** in the **X**, **Y**, and **Z** edit fields as the length for all three dimensions.
- 4 Click **OK** to create the block.

Block				X
Styl () ()	e Solid Face	Base Corner Center	Axis base point x: 0 y: 0 z: 0	Rotation angle a: 0 (degrees)
X: Y: Z: Name	gth 2 2 2 2		Axis direction vector Cartesian coordinates x: 0 y: 0 z: 1	Spherical coordinates θ: 0 (degrees) φ: 0 (degrees)
			OK Cancel	Apply Help

5 To ensure that the full geometry is visible, click the Zoom Extents button.

Now move on to the two cylinders. To create these 3D objects on a 2D screen, work in a 2D work plane. The method to follow in creating the cylinders is simple: draw two circles in 2D and then extrude them in the z direction to form 3D objects.

- I Go to the Draw menu and choose Work-Plane Settings to define a new work plane.
- 2 Click the Quick tab and make sure that the x-y button is selected (the default).
- **3** Type -0.2 in the **z** edit field.
- 4 Click **OK** to enter the 2D work plane.

Now draw the cylinder cross sections in the work plane:

- I Click on the **Zoom In** button several times until the grid spacing equals 0.2 in both the *x* and *y* directions.
- 2 Go to the Options menu and choose Axes/Grid Settings.
- 3 In the Axes/Grid Settings dialog box click on the Grid tab.
- 4 Clear the Auto check box and type 0.1 in the y spacing edit field.
- **5** Click **OK**. You can now create circles with a radius of 0.1.
- 6 Click on the Ellipse/Circle (centered) button in the Draw toolbar.
- 7 Place the crosshairs at the coordinates (-0.4, 0). Then click on the right mouse button and drag up or down until you create a circle with radius of 0.1.
- 8 Repeat the process to draw a second circle of the same size centered at (0.4, 0).

Create the cylinders:

- I Press Ctrl+A to select both circles.
- 2 Go to the Draw menu and choose Extrude.
- 3 In the **Extrude** dialog box define the height of the cylinders by typing 0.4 in the **Distance** edit field. The resulting cylinders extend from −0.2 to 0.2 in the *z* direction.

Extrude		×
Objects to extrude:	Extrusion parame	ters
C1	Distance:	0.4
C2	Scale x:	1
	Scale y:	1
	Displacement x:	0
	Displacement y:	0
-	Twist (degrees):	0
Extrude to geometry:	V Keep cross-se	ectional boundaries
Geom1 👻		
Extruded object name:		
EXT1		
	ОКС	ancel Help

4 Click **OK** to close the dialog box.

COMSOL Multiphysics now returns to the 3D environment where the cylinders are present. The cylinders have become extruded geometry objects with the names EXT1 and EXT2.

To complete the geometry, subtract the two cylinders from within the box using a Boolean operation:

- I Click the Create Composite Object button in the Draw toolbar.
- 2 Type the Boolean expression BLK1-EXT1-EXT2 in the Set formula edit field.
- 3 Click OK.

The resulting geometry is a new composite object with the default label CO1.

PHYSICS MODELING

The next task is to define the physics through material properties on the domain and conditions on the boundaries.

Subdomain Settings

First, specify the material properties in the modeling domain.

Open the Subdomain Settings dialog box from the Physics menu. COMSOL

Multiphysics then switches to Subdomain Selection mode where you can specify material properties for each subdomain.

Subdomain Settings - Electro	statics (es)				X
Equation					
$-\nabla{\cdot}\epsilon_0\epsilon_r\nabla V=\rho$					
Subdomains Groups	Physics Init El	ement Color			
Subdomain selection	Material propert	ties and sources			
1	Library material	l: 🔶 🗸	Load		
	Constitutive	relation			
	• $\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}$	$\bigcirc \mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$	🔘 D =	$\epsilon_0 \epsilon_r \mathbf{E} + \mathbf{D}_r$	
	Quantity	Value/Expression	Unit	Description	
-	ρ	0	C/m ³	Space charge density	
Group:	٤٢	1		Relative permittivity	
Select by group					
Active in this domain					
	L				
		ОК	Cancel	Apply Hel	p

In this example, ε equals the permittivity of free space $\varepsilon_0 = 8.854 \cdot 10^{12}$ F/m, a value predefined in COMSOL Multiphysics. The relative permittivity ε_r then equals 1.

SETTINGS	SUBDOMAIN I
ε _r	1
ρ	0

The space charge density option specifies the distribution of a space charge in a subdomain. Here there is none, so the value of ρ equals zero.

Now set the value for the permittivity in the model:

- I This model consists of only one subdomain. Make certain that Subdomain 1 is selected in the **Subdomain selection** list.
- **2** Go to **Constitutive relation** and make certain that the default $\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}$ is selected.
- 3 Click OK.

For time-dependent and nonlinear analyses you might find it necessary to set initial conditions. That step is not necessary for this linear electrostatics problem.

Boundary Conditions

To set boundary conditions, go to the **Physics** menu and choose **Boundary Settings** to open up the corresponding dialog box. Here you select boundaries and specify the boundary conditions.

When you select a specific boundary, the program allows you to choose from a list of available boundary conditions.

Boundary Settings - Electrost	atics (es)	X
Equation		
V = 0		
Boundaries Groups	Conditions Color:	
Boundary selection	Boundary sources and constraints	5
1 ^ 2 E 3 E 4 5 6 7 Group: Select by group Interior boundaries	Boundary condition: Ground	
	OK Cancel Apply Help	

To speed up the modeling process even with a relatively simple 3D model, you can select several faces with one mouse click. To learn more about the various selection methods see "Object Selection Methods in 3D" on page 127 in the *COMSOL Multiphysics User's Guide*.

The following table summarizes the boundaries in this model and their values:

SETTINGS	BOUNDARIES 6-11	BOUNDARIES 12-17	BOUNDARIES 1, 2, 4, 5, 18	BOUNDARY 3
Туре	Electric potential	Electric potential	Zero charge/Symmetry	Ground
V_0	1	- 1		

• Boundaries 6–11 define the left cylinder where the electric potential is 1 V.

- The second cylinder consists of Boundaries 12–17 and has an electric potential of –1 V.
- The third set of boundaries defines the box, except the bottom boundary.
- The last set (Boundary 3) is the grounded bottom plate.

To enter these boundary values into the model, follow these steps:

- I To enable object selection with the mouse you must first deselect the orbit/pan/ zoom function. To do so, click the **Orbit/Pan/Zoom** button.
- 2 Draw a rubber-band box around one of the cylinders.
- **3** When its faces are highlighted, select **Electric potential** from the **Boundary condition** list.
- 4 Type 1 in the V₀ edit field.
- **5** Repeat the procedure for the second cylinder, but this time type -1 in the V_0 edit field.
- 6 Select the Select by group check box.
- 7 Select Boundary 1. The group selection then also selects all other exterior boundaries.
- 8 Select Symmetry from the Boundary condition list.
- **9** To select the bottom face, first click to clear the **Select by group** check box and then select Boundary **3**.

10 Select Ground from the Boundary condition list.

II Click OK.

MESH GENERATION

Generating the mesh is usually quite simple: just click one of the mesh buttons on the Main toolbar. In cases with special requirements you can also use commands in the **Mesh** menu. In this case, use a predefined setting for creating a coarse mesh, which makes the solution process faster than the default mesh-size settings.

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 In the Free Mesh Parameters dialog box, select Coarser from the Predefined mesh sizes list.
- 3 Click OK.
- 4 Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

At this point you can either accept the default settings for the solver COMSOL Multiphysics should use, or you can interactively select the solver and its operating parameters. In most cases, though, you only need to click on the **Solve** button in the Main toolbar.

To reduce the time and memory requirements for the solution process, it is wise to make a few changes to the solver parameters. The default settings for 3D electrostatics models specify an *iterative solver*. These solvers generally use less memory than direct solvers and are often faster.

Iterative solvers work with a set of *preconditioners*, and the *algebraic multigrid* (*AMG*) preconditioner works best for scalar elliptic PDEs such as Laplace's equation and Poisson's equations. It is the default preconditioner for the 3D Electrostatics application mode.

- I From the Solve menu, open the Solver Parameters dialog box.
- 2 Click the General tab.
- 3 Choose a linear system solver. Select GMRES from the Linear system solver list.
- 4 Choose a preconditioner. Select Algebraic multigrid from the Preconditioner list.
- 5 Click OK.

Analysis:	Gener	al Stationary A	daptive Advanced	
Auto select solver	Line	ar system solver	1	-
Solver:	Line	ar system solver:	GMRES	
Stationary	Prec	onditioner:	Algebraic multigrid	
Time dependent Eigenvalue	Qua	lity of multigrid hi	erarchy: 3	
Parametric Stationary segregated Parametric segregated	Mer	nory efficiency	Precond. quality	Settings
Parametric segregated	Mate	rix symmetry:	Automatic	
Adaptive mesh refinen	nent			

6 Start the solution process by clicking on the **Solve** button on the Main toolbar. The results might take a few seconds to compute depending on the problem's complexity.

POSTPROCESSING AND VISUALIZATION

After solving the problem, COMSOL Multiphysics shifts to its postprocessing mode and shows a slice plot of the electric potential V. This plot, however, might not offer much immediately useful information, but COMSOL Multiphysics provides a variety of color plots and animations that help you more quickly spot trends and regions of interest. To better examine the solution for an electrostatic problem, for instance, you might want to view a surface or contour plot, and as the variable of interest you can select the electrostatic potential, the electric field, the electric displacement field, or the polarization field. Further, for arrow or streamline plots it is possible to select the x, y, or z components of most of those parameters.

To combine a surface plot with a flow plot, follow these steps:

- I Choose Plot Parameters from the Postprocessing menu.
- 2 On the General page, clear the Slice check box to deactivate the slice plot.
- 3 Select the Boundary and Streamline check boxes to activate these features.
- 4 Click the **Streamline** tab.
- 5 Type 30 in the Number of start points edit field on the Start Points tab.
- 6 Click OK.

You should immediately notice that the surface plot obstructs the view, making it impossible to see inside the box. To remove the rendering of the front faces, use the ability of COMSOL Multiphysics to suppress boundaries.

- I In the Options menu, point to Suppress and then click Suppress Boundaries.
- **2** While holding down the Ctrl key, select Boundaries 1, 2, and 4 in the **Boundary** selection list.
- 3 Click **OK** to close the dialog box and render all boundaries except 1, 2, and 4.



4 To add lighting to the model, click the **Scene Light** and **Headlight** buttons on the Camera toolbar.

Pacemaker Electrode

This model illustrates the use of COMSOL Multiphysics for modeling of ionic current distribution problems in electrolytes, in this case in human tissue. The problem is exemplified on a pacemaker electrode, but it can be applied in electrochemical cells like fuel cells, batteries, corrosion protection, or any other process where ionic conduction takes place in the absence of concentration gradients.

Introduction

The modeled device is a pacemaker electrode that is placed inside the heart and helps the patient's heart to keep a normal rhythm. The device is referred to as an electrode, but it actually consists of two electrodes: a cathode and an anode.

Figure 4-2 shows a schematic drawing of two pair of electrodes placed inside the heart. The electrodes are supplied with current from the pulse generator unit, which is also implanted in the patient.



Figure 4-2: Schematic drawing of the heart with two pairs of pacemaker electrodes.

This model deals with the current and potential distribution around one pair of electrodes.

Model Definition

The model domain consists of the blood and tissue surrounding the electrode pair. The actual electrodes and the electrode support are boundaries to the modeled domain. Figure 4-3 shows the electrode in a darker shade, while the surrounding modeling domain is shown in a lighter shade.



Figure 4-3: Modeling domain and boundaries.

The working electrode consists of a a hemisphere placed on the tip of the supporting cylindrical structure. The counter electrode is placed in the "waist" of this structure. All other surfaces of the supporting structure are insulated. The outer boundaries are placed far enough from the electrode to give a small impact on the current and potential distribution.

In COMSOL Multiphysics, use the 3D Conductive Media DC application mode for the analysis of the electrode. This application mode is useful for modeling conductive materials where a current flows due to an applied electric field.

DOMAIN EQUATIONS

The current in the domain is controlled by the continuity equation, which follows from the Maxwell's equations:

$$-\nabla \cdot (\sigma \nabla V) = 0$$

where σ is the conductivity of the human heart. This equation uses the following relations between the electric potential and the fields.

 $\mathbf{E} = -\nabla V$ $\mathbf{J} = \mathbf{\sigma} \mathbf{E}$

BOUNDARY CONDITIONS

Ground potential boundary conditions are applied on the thinner waist of the electrode. The tip of the electrode has a fixed potential of **1** V. All other boundaries are electrically insulated.

$$\mathbf{n} \cdot \mathbf{J} = \mathbf{0}$$

Results and Discussion

This simulation gives you the potential distribution on the surface of the electrode and the streamlines of the current distribution inside the human heart. This plot appears in Figure 4-4.



Figure 4-4: The plot shows the electrostatic potential distributed on the surface of the electrode. The total current density is shown as streamlines.

As expected, the current density is highest at the small hemisphere, which is the one that causes the excitation of the heart. The current density is fairly uniform on the

working electrode. The counter electrode is larger and there are also larger variations in current density on its surface. Mainly, the current is lower with the distance from the working electrode. The model shows that the anchoring arms of the device have little influence on the current density distribution.

Model Library path: COMSOL_Multiphysics/Electromagnetics/

pacemaker_electrode

Modeling Using the Graphical User Interface

- I In the Model Navigator, select 3D from the Space dimension list.
- 2 In the COMSOL Multiphysics>Electromagnetics folder, select Conductive Media DC. Make sure Lagrange - Quadratic is selected in the Element list.
- 3 Click OK.

GEOMETRY MODELING

- I Select Work-Plane Settings from the Draw menu.
- 2 Click the y-z button and then click OK.
- 3 From the Draw menu, select Specify Objects>Rectangle. In the Rectangle dialog box, enter 2.1e-3 in the Width edit field, 5.5e-3 in the Height edit field, and enter the corner coordinates (0, 14.5e-3) in the edit fields x and y. Make sure that Corner is selected in the Base list. Click OK.
- 4 Select Fillet/Chamfer from the Draw menu. Expand R1 and select point number 2. Enter 5e-4 in the Radius edit field. Click OK.
- 5 Create a rectangle with width 1.6e-3, height 2e-3, and corner position at (0, 12.5e-3).
- 6 Create a rectangle with width 2.1e-3, height 12.5e-3, and corner position at (0, 0).
- 7 Select Fillet/Chamfer from the Draw menu. Expand R2 and select point number 2 and3. Enter 5e-4 in the Radius edit field. Click OK.
- 8 From the Draw menu, select Specify Objects>Circle. Enter 1e-3 in the Radius edit field. Click OK.
- 9 Create a rectangle with width 2e-3, height 2e-3, and corner position at (-2e-3, -1e-3).

- **10** Select **Create Composite Object** from the **Draw** menu. In the dialog box, type the formula C1-R2 in the **Set formula** edit field. Click **OK**.
- II Select all objects by pressing Ctrl+A. Click on the Union toolbar button located to the left side.
- 12 Click the Delete Interior Boundaries toolbar button on the same toolbar.
- **13** Click the **Zoom Extents** button on the Main toolbar. You should have a window similar to the figure below.



14 Select Revolve from the Draw menu. Click OK.

IS Click the Zoom Extents button to see the revolved geometry.

These steps created the electrode, but the electrode also has some hooks that hold it in place.

- I Go back to the work plane by clicking on the **Geom2** tab.
- 2 Create a rectangle with width 5e-4, height 5.2e-3, and corner position at (0, 3.5e-3).
- 3 Select Fillet/Chamfer from the Draw menu. Expand R1 and select point number 3. Click the Chamfer radio button, and enter 2e-4 in the Distance edit field. Click OK.
- 4 Select Revolve from the Draw menu. Click OK.

- 5 Click the Rotate toolbar button and enter 60 in the edit field for the rotation angle. Define the point and axis for the rotation by specifying the coordinate (0, 0, 2.5e-3) in the Point on rotation axis frame, and the direction (1, 0, 0) in the Rotation axis direction vector frame. Click OK.
- 6 Press Ctrl+C to copy the object. Press Ctrl+V to paste it directly. In the **Paste** dialog box, leave all displacement fields set to zero. Click **OK**.
- 7 Click the Rotate toolbar button and set the rotation angle to 90. Click OK.
- 8 Press Ctrl+V again and click OK (zero displacements).
- 9 Click the Rotate toolbar button and set the rotation angle to 180. Click OK.
- **IO** Press Ctrl+V and click **OK** (zero displacements).
- II Click on the Rotate toolbar button and set the rotation angle to 270. Click OK.
- **12** Press Ctrl+A to select all objects. Click on the **Union** toolbar button and then click on the **Delete Interior Boundaries** toolbar button.
- **13** Click the **Zoom Extents** and the **Headlight** toolbar buttons to see geometry in the following figure.



Finally, you need to define the volume surrounding the electrode. The simulation only takes place in this volume, where the boundaries of the electrode influence the result.

- Click the Cylinder toolbar button. In the Cylinder dialog box, enter 10e-3 in the Radius edit field, 40e-3 in the Height edit field, and the coordinate (0, 0, -20e-3) in the edit fields of the Axis base point frame. Click OK.
- 2 Choose Create Composite Object from the Draw menu. In the dialog box, type the formula CYL1-CO2 in the Set formula edit field. Click OK.

PHYSICS SETTINGS

Boundary Conditions

The thin waist of the electrode is grounded, and a positive potential is applied to the lower half sphere. All other boundaries are kept electrically insulated.

I Open the Boundary Settings dialog box from the Physics menu.

2 Define the boundary condition according to the following table. Click OK.

SETTINGS	BOUNDARIES 29, 30, 58, 63	BOUNDARIES 31, 32, 59, 60	ALL OTHER BOUNDARIES
Boundary condition	Ground	Electric potential	Electric insulation
V ₀		1	

Subdomain Settings

The electrode is inserted into the human heart, so you must define the conductivity for the heart tissue.

- I Open the Subdomain Settings dialog box from the Physics menu.
- 2 Enter the following values in the corresponding edit fields. Click OK.

SETTING	SUBDOMAIN I
σ (isotropic)	5000

MESH GENERATION

Use the default mesh settings. Click the **Initialize Mesh** button on the Main toolbar to generate the mesh.

COMPUTING THE SOLUTION

Use the default solver parameters—the stationary solver using the conjugate gradients iterative solver—and algebraic multigrid as the preconditioner.

Click on the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The potential distribution on the electrode boundary is used together with a streamline plot of the total current density to visualize the result of the simulation.

- I Start by suppressing some boundaries not to be used for visualization. Select Suppress Boundaries from the Options menu, and select Boundaries 1, 2, 3, 4, 45, and 74. Click OK.
- 2 Choose Plot Parameters from the Postprocessing menu. In the Plot Parameters dialog box under the General tab, clear the Slice check box and select the Boundary and Streamline check boxes.
- **3** Click the **Boundary** tab and make sure that **Electric potential** is selected as boundary data.
- 4 Click the **Streamline** tab and then select **Total current density** from the **Predefined quantities** list. Click the **Line Color** tab, and then click the **Use expression** radio button. Use the default expression—the electric potential.
- 5 Select Tube from the Line type list. Click the Tube Radius button. In the Tube Radius Parameters dialog box, select the Radius data check box, and choose Total current density, norm from the Predefined quantities list. Click OK twice. You should now see something similar to the figure below after proper rotation and zoom operations.



Skin Effect in Circular Wire

This example demonstrates the *skin effect* where, during the flow of AC current, electrons tend to move along the surface of a conductor. Changes in the current's amplitude and direction induces a magnetic field that pushes the electrons toward the wire's exterior. The effect increases with frequency and conductor size.

Engineers working at microwave frequencies take advantage of these effects by designing hollow waveguides because at these frequencies, the core of a conductor would not carry current anyway.

In solid conductors, the skin effect can have important implications even at powerline frequencies. For instance, utilities can replace expensive, heavy copper wire with aluminum cables clad with a copper skin without appreciably increasing power losses. As proof of this claim, this model first computes the current distribution at 50 Hertz in an unusually large copper cable (20 cm diameter). It then goes on to compute the current distribution and compare resistive losses in a like-sized copper-clad cable that consists of 90% aluminum.

Model Definition

Copper specifies a conductivity σ of 5.99 $\cdot 10^7$ S/m and permeability μ of $4\pi \cdot 10^{-7}$ H/m. For aluminum the values are $3.77 \cdot 10^7$ S/m and $4\pi \cdot 10^{-7}$ H/m, respectively. The materials' dielectric properties have no influence on the fields at low frequencies as in this case. You can therefore set $\varepsilon = \varepsilon_0 = 8.854 \cdot 10^{-12}$ F/m.

Using the AC Power Electromagnetics application mode, the equation that COMSOL Multiphysics solves is a complex Helmholtz equation for the amplitude of the magnetic potential:

$$-\nabla \cdot \left(\frac{1}{\mu}\nabla A_z\right) + k^2 A_z = 0$$
$$k = \sqrt{j\omega\sigma - \omega^2 \varepsilon}$$

Here ω is the angular frequency, in this case $2\pi \cdot 50$ rad/s. For good conductors, such as copper and aluminum, $\sigma \gg \omega \epsilon$. So, for engineering purposes the approximation $k = \sqrt{j\omega\sigma}$ is close enough.

The boundary conditions require some careful consideration. From a mathematical point of view, it is necessary to specify either the magnetic potential A_z (corresponding to a Dirichlet type condition) or the normal derivative of the same field (a Neumann condition) on the outer surface. These quantities, however, have little or no significance in applied engineering. In COMSOL Multiphysics the Neumann type condition on the A_z field is implemented by specifying a surface current equal to the negative of the tangential magnetic field at the boundary. The current may not be physically real. On exterior boundaries you can interpret it as the surface current necessary to make the magnetic field **H** vanish outside the domain.

Even better from the pure engineering point of view is to specify the total current throughput, which is rather straightforward in this case. Making the magnetic field disappear everywhere outside the circular domain requires that the total current through the domain equals zero. One way of achieving this is by adding a "virtual" surface current of opposite sign to the "real" current inside the conductor. Because the problem is rotationally symmetric, you can write the necessary virtual surface current density

$$J_s = -\frac{I_{\text{tot}}}{2\pi R}$$

where I_{tot} is the total real current throughput, and R is the radius of the wire.

Results



The solution of the AC power electromagnetics equation is complex valued, reflecting the time lag between the surface and the interior. This plot shows the real part of the solution. The full complex-valued solution is available. You can plot various properties of the complex solution by typing expressions such as imag(Jz_qa) or abs(Jz_qa) in an **Expression** edit field in the **Plot Parameters** dialog box.

Model Library path: COMSOL_Multiphysics/Electromagnetics/skin_effect

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select 2D in the Space dimension list.
- 2 In the list of application modes, open the **COMSOL Multiphysics** folder and then the **Electromagnetics** folder. Select **AC Power Electromagnetics** from the list of electromagnetics application modes.

- **3** Accept the default element type (quadratic Lagrange elements), which is suitable for most cases.
- 4 Click OK.

OPTIONS AND SETTINGS

- I From the Options menu, choose Axes/Grid Settings.
- 2 Set axis and grid settings according to the following table. On the **Grid** page, clear the **Auto** check box to enter the grid spacings manually.

AXIS		GRID	
x min	-0.15	x spacing	0.1
x max	0.15	Extra x	
y min	-0.1	y spacing	0.1
y max	0.1	Extra y	

- 3 Click OK.
- 4 From the **Options** menu, choose **Constants**.
- **5** Enter the following constants:

NAME	EXPRESSION
sigC	5.99e7
sigA	3.77e7
Itot	100

6 Click OK.

GEOMETRY MODELING

The cross-section of the conductor is represented by a circle with radius 0.1. The aluminum core is drawn as a circle with radius 0.0949.

- I Draw a circle centered at (0, 0) with a radius of 0.1.
- **2** Draw one more circle centered at (0, 0) with a radius of 0.1.
- **3** Double-click the last circle and enter a new radius of 0.0949.

PHYSICS SETTINGS

Boundary Conditions

I From the Physics menu, choose Boundary Settings.

2 Enter boundary coefficients according to the following table. When done, click **OK**.

BOUNDARY	ALL
Туре	Surface current
J_{sz}	-Itot/(2*pi*0.1)

Subdomain Settings

- I From the Physics menu, choose Subdomain Settings.
- **2** Click the **Electric Parameters** tab.
- **3** Enter the subdomain settings (material property) according to the following table. When done, click **OK**.

SETTING	SUBDOMAINS I, 2
σ	sigC

MESH GENERATION

The small gap between the concentric circles gives you a fine mesh close to the boundaries, where the solution is expected to vary fastest. If you decrease the element growth rate, the mesh becomes smoother.

I In the Free Mesh Parameters dialog box, click the Custom mesh size button and type1.1 in the Element growth rate edit field.

2 Initialize the mesh.



COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

Due to the skin effect, the current density at the surface is much higher than that within the interior of the conductor. Plotting the current density J_z clearly shows this effect.

- I Open the Plot Parameters dialog box.
- **2** Click the **Surface** tab.
- **3** On the Surface Data tab, select Total current density, z component from the Predefined quantities list.
- 4 Select cool from the Colormap list.
- 5 Click OK.

The resistance per meter is defined as $R = P/I^2$, where *P* is the power loss per meter of wire, and *I* is the current through the power line. You can compute both the power and the current with integrations over the cross section:

$$P = \int Q dA$$
$$I = \int J_z dA$$

where $Q = \sigma |\mathbf{E}|^2$. The value of the second integral is known, because the total current is part of the boundary conditions. However, performing the integration of the FEM solution provides an opportunity to verify that the previous calculations are correct.

- I Open the Subdomain Integration dialog box. Select both subdomains and choose the Total current density, z component from the Predefined quantities list. Click OK and note that the total current is very close to 100 A, as expected.
- 2 Change the integration expression to **Resistive heating, time average** and again click **OK**.
- **3** You can now compute the resistance per meter from $R = P/I^2$.

Modeling a Wire with an Aluminum Core

Next, replace almost all copper with aluminum, leaving only a thin copper shell, and compare the results.

SUBDOMAIN SETTINGS

Change the conductivity in Subdomain 2 to sigA.

COMPUTING THE SOLUTION

Solve the problem.

POSTPROCESSING

Now calculate the resistance per meter in the modified design.

- I Open the Subdomain Integration dialog box.
- 2 Select both subdomains and choose **Resistive heating**, time average in the **Predefined** quantities list.
- 3 Click OK.
- **4** Calculate the resistance per meter as before.

For a given current you can see that the combined copper/aluminum wire suffers approximately 10 percent more resistance compared to pure copper. On the other hand it weighs only one third and is considerably less expensive.

I To solve the problem using the application mode, first set some constants, define the geometry and create the mesh:

```
clear fem
omega = 2*pi*50;
Itot = 100;
fem.geom = circ2(0,0,0.1)+circ2(0,0,0.0949);
```

2 Select the AC Power Electromagnetics application mode:

```
fem.appl.mode = 'FlPerpendicularCurrents';
fem.appl.prop.analysis = 'harmonic';
```

3 Specify the boundary conditions:

fem.appl.bnd.JsOz = -Itot/(2*pi*0.1); fem.appl.bnd.type = {'Js'};

4 Specify the material property:

```
fem.appl.equ.sigma = 5.99e7;
```

5 Expand the application mode fem.app1, mesh the geometry, and create the fem.xmesh field:

```
fem = multiphysics(fem);
fem.mesh = meshinit(fem, 'hgrad', 1.1);
fem.xmesh = meshextend(fem);
```

6 Solve the problem with the stationary solver:

```
fem.sol = femstatic(fem);
```

7 Visualize the result by plotting the current density:

postplot(fem,'tridata','Jz','trimap',cool); axis equal

8 Integrate the power loss and the current over the cross-section and calculate the resistance per meter of wire:

```
P1 = postint(fem,'Qav')
I1 = postint(fem,'Jz')
R1 = P1/abs(I1)^2
```

9 Change the conductivity of the core, rebuild the FEM structure, and solve again:

```
fem.appl.equ.sigma = {5.99e7 3.77e7};
fem = multiphysics(fem);
fem.xmesh = meshextend(fem);
fem.sol = femstatic(fem);
```

IO Plot the current distribution in a new window:

figure;

postplot(fem,'tridata','Jz','trimap',cool); axis equal

II Perform the integrations over the cross section and calculate the resistance:

P2 = postint(fem, 'Qav')
I2 = postint(fem, 'Jz')
R2 = P2/abs(I2)^2

12 Finally calculate the ratio of the resistances in the two cases:

Q = R2/R1

Quadrupole Lens

Introduction

Just like optical lenses focus light, electric and magnetic lenses can focus beams of charged particles. Systems of magnetic quadrupole lenses find a common use in focusing both ion and particle beams in accelerators at nuclear and particle physics centers such as CERN, SLAC, and ISIS. This COMSOL Multiphysics model shows the path of Be^{5+} ions going through three consecutive magnetic quadrupole lenses. The model is set up in a cross section of the geometry. You can find a full 3D version in the AC/DC Module Model Library.

Model Definition

The quadrupole consists of an assembly of four permanent magnets, as seen in Figure 4-5 below, where the magnets work together to give a good approximation of a quadrupole field. To strengthen the field and keep it contained within the system, the magnets are set in an iron cylinder.



Figure 4-5: Cross-sectional view of one of the magnetic quadrupoles used in the lens.

The ions are sent through a system of three consecutive quadrupole assemblies. The middle one is twice as long as the other ones, and is rotated by 90 degrees around the central axis. This means the polarity of its magnets is reversed. Figure 4-6 gives a full view of the magnetic quadrupole lens.



Figure 4-6: Cutout of the quadrupole lens. The second quadrupole (Q2) has its polarities reversed compared to Q1 and Q2. After traveling through the lens, the ions are left to drift 1 m.

An accelerator feeds the system with ions traveling with the velocity 0.01*c* along the the central axis. To study the focusing effect of the quadrupoles, track a number of ions starting out from a distance of 3 cm from the central axis, evenly distributed along the circumference of a circle in the transverse plane. They are all assumed to have a zero initial transverse velocity. Each quadrupole focuses the ion beam along one of the transverse axes and defocuses it along the other one. The net effect after traveling through the system of the three quadrupoles and the drift length is focusing in all directions. As the ions exit the system, they are all contained within a 1 cm radius in the transverse plane.

The model is set up in a 2D cross section of any of the two identical quadrupoles Q1 and Q3. Neglecting fringe fields, the transverse magnetic field at a given point in a transverse plane in Q2 will automatically have the same magnitude as the corresponding in Q1 and Q3, but point in the opposite direction. It is therefore sufficient to model the fields in one of the quadrupoles.

DOMAIN EQUATIONS

The magnetic field is described using the Magnetostatics equation, solving for the *z* component of the magnetic potential \mathbf{A} (Wb/m):

$$\nabla \times (\mu_0^{-1} \nabla \times A_z - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times A_z) = \sigma \Delta V / L + J_z^{e}$$

Here $\mu_0 = 4\pi \cdot 10^{-7}$ H/m denotes the permeability of vacuum, **M** is the magnetization (A/m), σ the conductivity (S/m), and **v** the velocity of the medium (m/s). In this model, the medium is not moving. The right-hand side of the equation holds an imposed current, specified in terms of either ΔV (V) and L (m), or an external current density $J_z^{\ e}$ (A/m²). No currents are imposed in this model. The iron subdomain uses a slightly different formulation of the same equation:

$$\left(\nabla \times \frac{1}{\mu_0 \mu_r} \nabla \times A_z\right) - \sigma \mathbf{v} \times (\nabla \times A_z) = \sigma \Delta V / L + J_z^e,$$

where $\mu_r = 4000$ is the relative permeability. The magnetic potential is everywhere defined so that $\mathbf{B} = \nabla \times \mathbf{A}$.

BOUNDARY CONDITIONS

The magnetic field is approximately parallel to the exterior boundary of the iron cylinder. To enforce this, use the *magnetic insulation* boundary condition, stating that $A_z = 0$.

Results



The magnetic field density and flowlines in a cross section of Q1 or Q3 appear in Figure 4-7 below.

Figure 4-7: The magnetic field density and flowlines in the center of one of the quadrupole magnets.

Each ion passing through the assembly experiences a Maxwell force equal to $\mathbf{F} = q \mathbf{v} \times \mathbf{B}$, where \mathbf{v} (m/s) is the ion's velocity. Next, assume that the *z* component of the velocity is constant and much larger than the *x* and *y* (transverse) components. Thus consider only the force contributions from the *z* component of the velocity. To find the transverse position as a function of time, you need to solve Newton's second law for each ion, $q\mathbf{v} \times \mathbf{B} = m\mathbf{a}$, where *m* is the ion mass (kg), and **a** denotes its acceleration (m/s²). If the computed magnetic flux density in Q1 equals **B**', and the length of quadrupole *i* is L_i (m), the flux density that the ion experiences is given by

$$\mathbf{B} = \begin{cases} \mathbf{B}' \text{ if } t < \frac{L_1}{v_z} \\ -\mathbf{B}' \text{ if } \frac{L_1}{v_z} < t < \frac{L_1 + L_2}{v_z} \\ \mathbf{B}' \text{ if } \frac{L_1 + L_2}{v_z} < t < \frac{L_1 + L_2 + L_3}{v_z} \\ 0 \text{ if } t > \frac{L_1 + L_2 + L_3}{v_z} \end{cases}$$

where t (s) is the time of flight. This dependency of the magnetic flux density on the time of flight is fed to the particle-tracing algorithm as a logical expression. Figure 4-8 below shows how the ions travel in the transverse plane.



Figure 4-8: As the ions enter Q1, they start out evenly distributed around the larger circle, 3 cm from the z-axis. Q1 focuses along the x-axis and defocuses along the y-axis. The force on each ion is approximately proportional to its distance from the z-axis, so as the ions enter Q2, those that are far out on the x-axis rapidly turn around and move toward the center. Q3 stabilizes the motion and gets all ions on the right track. Finally the ions are left to drift toward a waist situated a little bit more than 1 m beyond Q3.

Model Library path: COMSOL_Multiphysics/Electromagnetics/quadrupole

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select COMSOL Multiphysics>Electromagnetics>Magnetostatics from the list of application modes.
- 2 Click **OK** to close the dialog box.

OPTIONS AND SETTINGS

I Enter the following constant names and expressions in the **Constants** dialog box (units and descriptions are optional):

NAME	EXPRESSION	DESCRIPTION
М	11	Ion mass number
Z	5	Ion charge number
L1	1[m]	Length of first quadrupole
L2	2[m]	Length of second quadrupole
L3	1[m]	Length of third quadrupole
vz	0.01*3e8[m/s]	Ion velocity
mp	1.672e-27[kg]	Proton mass
Ze	1.602e-19[C]	Proton charge
m	M*mp	Ion mass
q	Z*Ze	Ion charge
MQ	5.8e3[A/m]	Quadrupole magnetization

2 Click **OK** to close the dialog box.

GEOMETRY MODELING

I Shift-click the Rectangle/Square button on the Draw toolbar.

2 Specify a rectangle with the following properties; when done, click **OK**.

PROPERTY	EXPRESSION
Width	0.177
Height	0.07

PROPERTY	EXPRESSION	
Position: Base	Corner	
Position: x	0	
Position: y	-0.035	

- **3** Click the **Rotate** button on the Draw toolbar. Rotate the rectangle by 45 degrees around the origin.
- 4 Click the Ellipse/Circle (Centered) button on the Draw toolbar.
- 5 Specify a circle with the following properties; when done, click **OK**.

EXPRESSION
0.2
Center
0.2
0.2

- **6** Select both the circle and the rectangle, then click the **Intersection** button on the Draw toolbar.
- 7 Make a copy of your composed object (CO1) and paste it at the same location, that is with zero displacements.
- 8 Rotate the copied object (CO2) by 90 degrees around the origin.
- **9** Paste two more copies of CO1 at the same location, and rotate them by 180 and 270 degrees respectively.
- **IO** Create a circle centered at the origin with a radius of 0.2 m.
- II Create another circle centered at the origin, with a radius of 0.12 m.
- 12 Click the Create Composite Object button and create an object using the formula C1+C2- (C01+C02+C03+C04).
- **I3** Draw a circle centered at the origin with a radius of 0.2 m.

The geometry should now look like that in the figure below.



Figure 4-9: The COMSOL Multiphysics geometry of the quadrupole lens.

PHYSICS SETTINGS

Boundary Conditions

Use the default condition Magnetic insulation at all exterior boundaries.

Subdomain Settings

- I From the Physics menu, select Subdomain Settings. Select Subdomain 1.
- 2 Click the Load button. In the Materials/Coefficients Library dialog box that appears, select Basic Material Properties>Iron in the Materials area.
- **3** Leave the default settings in Subdomain 4.
- **4** In all the other subdomains, set the constitutive relation to $\mathbf{B} = \mu_0 \mathbf{H} + \mu_0 \mathbf{M}$ and specify the magnetization components according to the table below.

PROPERTY	SUBDOMAIN 2	SUBDOMAIN 3	SUBDOMAIN 5	SUBDOMAIN 6
M (x component)	MQ/sqrt(2)	-MQ/sqrt(2)	MQ/sqrt(2)	-MQ/sqrt(2)
M (y component)	MQ/sqrt(2)	MQ/sqrt(2)	-MQ/sqrt(2)	-MQ/sqrt(2)

MESH GENERATION

I From the Mesh menu, select Free Mesh Parameters.

- 2 From the Predefined mesh sizes list, select Extra fine.
- 3 Click the **Remesh** button.
- 4 When the mesher has finished, click **OK** to close the dialog box.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The default plot shows the norm of the magnetic flux density. Follow the instructions to view the magnetic field.

- I Click the **Plot Parameters** button on the Main toolbar.
- 2 On the Surface page, select Magnetic field, norm from the Predefined quantities list.
- **3** On the **Contour** page, select the **Contour plot** check box. From the **Predefined quantities** list, select **Magnetic potential**, **z component**.
- 4 In the Contour color area, click first the Uniform color option button and then the Color button. Select a white color, then click OK. Clear the Color scale check box.
- **5** Click **Apply** to see the magnetic field. The magnetic field lines follow the same pattern as the magnetic potential contour lines (see Figure 4-7 on page 86).

To see how the ions travel through the system of quadrupoles, do the following:

- 6 On the Surface page, set the Colormap to hot.
- **7** On the **Particle Tracing** page, select the **Particle tracing plot** check box and apply the following settings:

PROPERTY	EXPRESSION
Mass	m
Fx	-q*vz*By_qa*(1-2*(partt>L1/vz)+2*(partt>(L1+L2)/vz) -(partt>(L1+L2+L3)/vz))
Fy	q*vz*Bx_qa*(1-2*(partt>L1/vz)+2*(partt>(L1+L2)/vz) -(partt>(L1+L2+L3)/vz))
Start Points, x	0.03*cos(linspace(0,2*pi,41))
Start Points, y	0.03*sin(linspace(0,2*pi,41))

- 8 On the Line Color page, click the Use expression option button.
- 9 Click the Color Expression button. Set the Colormap to cool and the expression to 1+(partt>L1/vz)+(partt>(L1+L2)/vz)+(partt>(L1+L2+L3)/vz). Click OK.

- 10 Click the Advanced button. Set the Relative tolerance to 1e-6, the End time for static flow fields to 5/3e6, and the Maximum number of steps to 1e5. Click OK.
- II On the Contour tab, set the Expression to $sqrt(x^2+y^2)$. For Contour levels, select Vector with isolevels and enter 0.01 0.03.
- 12 Click **OK** to generate the plot.
- **13** Click the **Zoom In** button on the Main toolbar to zoom in on the center of the model geometry as in Figure 4-8 on page 87.

Spherical Capacitor

Introduction

This example is a study of a capacitor consisting of two spherical copper conductors separated by glass.

You can compute the capacitance of a capacitor in two ways. By definition, the capacitance is given by the expression

$$C = \frac{Q}{V_{12}}$$

where V_{12} is the voltage difference between the two conductors and Q is the charge of the two conductors, positive and negative, respectively, when connected in a DC circuit.

The energy required to charge a capacitor is given by the expression

$$W_e = \frac{Q^2}{2C}$$

The energy defined above should equal the energy of the electrostatic field:

$$W_e = \int_{\Omega} \mathbf{D} \cdot \mathbf{E} \, dV$$

Because this integral is readily available in the Electrostatics application mode, the capacitance is obtained from the expression

$$C = \frac{Q^2}{2W_e}$$

With the second method, an analytical expression for the capacitance is calculated and used for a comparison with the numerical solution. Thus, assume charges +Q and -Q on the inner and outer spherical conductor, respectively. Denote the radii of the two spheres by R_i and R_o , respectively. Applying Gauss's law to a spherical surface with radius R ($R_i < R < R_o$) gives

$$\mathbf{E} = \frac{Q}{4\pi\epsilon R^2} \mathbf{e}_{z}$$

The potential difference between the spheres is then

$$V = -\int_{R_o}^{R_i} \mathbf{E} \cdot \mathbf{e}_r \, dR = \frac{Q}{4\pi\varepsilon} \left(\frac{1}{R_i} - \frac{1}{R_o}\right)$$

Hence, the analytical expression for the capacitance is

$$C = \frac{Q}{V} = 4\pi\varepsilon / \left(\frac{1}{R_{\rm i}} - \frac{1}{R_{\rm o}}\right)$$

Model Definition

In COMSOL Multiphysics, you solve the spherical capacitor problem using the axisymmetric 2D Electrostatics application mode. It is easy to compute the capacitance from the COMSOL Multiphysics user interface using the available integration tools.

DOMAIN EQUATIONS

The electric scalar potential V, must obey Poisson's equation,

$$-\nabla \cdot (\varepsilon_0 \varepsilon_r \nabla V) = \rho$$

where ε_0 is the permittivity of free space, ε_r is the relative permittivity and ρ is the space charge density. The electric field and displacement are obtained from the gradient of *V*

$$\mathbf{E} = -\nabla V$$
$$\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}$$

BOUNDARY CONDITIONS

A ground potential boundary condition describes the condition at the outer electrode. At the inner electrode, the boundary condition gives the surface charge:

$$-\mathbf{n} \cdot \mathbf{D} = \rho_s$$

A symmetry boundary condition describes the condition at the symmetry axis.

Model Library path: COMSOL_Multiphysics/Electromagnetics/ spherical_capacitor
MODEL NAVIGATOR

- I Select Axial symmetry (2D) from the Space dimension list.
- 2 In the list of application modes, select

COMSOL Multiphysics>Electromagnetics>Electrostatics.

3 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Axes/Grid Settings**.
- 2 Set axis and grid settings according to the following table. To enter the settings on the **Grid** page, first clear the **Auto** check box. When done, click **OK**.

AXIS		GRID	
r min	-0.2	r spacing	0.05
r max	0.3	Extra r	0.19
z min	-0.2	z spacing	0.05
z max	0.2	Extra z	

3 In the **Constants** dialog box, enter the following variable name and expression, and then click **OK**.

NAME	EXPRESSION
QO	5e-12

GEOMETRY MODELING

- I Draw a circle C1 with radius 0.19 centered at (0, 0).
- **2** Draw a circle C2 with radius 0.05 centered at (0, 0).
- **3** Draw a rectangle with opposite corners at (-0.2, -0.2) and (0, 0.2).
- **4** Open the **Create Composite Object** dialog box from the Draw toolbar or the **Draw** menu.
- 5 Form a composite object by typing C1 (C2+R1) in the Set formula edit field.

6 Click **OK** to close the dialog box, then click the **Zoom Extents** button on the Main toolbar.



PHYSICS SETTINGS

Boundary Conditions

From the **Physics** menu, choose **Boundary Settings**. Enter boundary conditions according to the following table and then click **OK**.

SETTINGS	BOUNDARIES I, 2	BOUNDARIES 4, 5	BOUNDARIES 3, 6
Туре	Axial symmetry	Surface charge	Ground
ρ_s		QO/(4*pi*0.05^2)	

Subdomain Settings

In this model, use the materials library for defining the material properties:

- I Open the **Subdomain Settings** dialog box, and select Subdomain 1.
- 2 Click the Load button to open the Materials/Coefficients Library dialog box.
- **3** Open the **Basic Material Properties** folder and select **Glass (quartz)** from the list of materials and click **OK**.
- 4 Now the defined material's relative permittivity is used in the subdomain settings.

MESH GENERATION

- I Initialize the mesh by clicking the Initialize Mesh button on the Main toolbar.
- 2 Click the **Refine Mesh** button on the Main toolbar to refine the mesh once.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The resulting electrostatic potential is zero at the outer conductor and positive and constant at the inner conductor as shown in the default surface plot of V.



Capacitance Calculation

You can compute the capacitance of the capacitor using the available postprocessing functionality in COMSOL Multiphysics:

- I Choose Boundary Integration from the Postprocessing menu and integrate the expression V/(0.05*pi) for Boundaries 4 and 5. This gives an average value of 0.157676 for the potential on the inner conductor.
- 2 Choose Subdomain Integration from the Postprocessing menu, select the Compute volume integral check box, and choose Electrostatics (es)>Electric energy density from the Predefined quantities list to compute volume integral of the electric energy

density, We_es. This gives a value of $3.94188 \cdot 10^{-13}$ joule for the total electric energy. It is now possible to compute the capacitance in two different ways:

C1 = Q/Vwhich gives the result C1 = 3.17106e-011and C2 = $Q^2/(2*We)$ which gives the result C2 = 3.17107e-11Compare this to the analytical result C0 = 4.2*4*pi*8.854e-12/(1/0.05-1/0.19)which gives the result C0 = 3.170985e-011

Clearly, both numerical results are in good agreement with the analytical expression.

Equation-Based Models

 $T_{\rm his}$ chapter contains models using the COMSOL Multiphysics PDE language to solve partial differential equations.

Electrical Signals in a Heart

This model is kindly provided by Prof. Simonetta Filippi and Dr. Christian Cherubini from Università Campus Biomedico di Roma, Italy.

Background

Modeling the electrical activity in cardiac tissue is an important step in understanding the patterns of contractions and dilations in the heart. The heart produces rhythmic electrical pulses, initiated from a point known as the sinus node. The electrical pulses, in turn, trigger the mechanical contractions of the muscle. In a healthy heart these electrical pulses are damped out, but a number of heart conditions involve an elevated risk of re-entry of the signals. This means that the normal steady pulse is disturbed, a severe and acute condition which is often referred to as arrhythmia.

This section presents two mathematical models describing different aspects of electrical signal propagation in cardiac tissue: the *FitzHugh-Nagumo* equations and the *Complex Ginzburg-Landau* equations, both of which are solved on the same geometry. Interesting patterns emerging from these types of models are, for example, spiral waves, which, in the context of cardiac electrical signals, can produce effects similar to those observed in cardiac arrhythmia.

EXCITABLE MEDIA AND THE FITZHUGH-NAGUMO EQUATIONS

It has been shown that many important characteristics of electrical signal propagation in cardiac tissue can be reproduced by a class of equations which describe *excitable media*, that is, materials consisting of elementary segments or cells with the following basic characteristics:

- Well-defined rest state
- · Threshold for excitation
- A diffusive-type coupling to its nearest neighbors

Excitable media is a rather general concept, which is useful for modeling of (in addition to the electrical signals in cardiac tissue) a number of different phenomena, including nerve pulses, the spreading of forest fires, and certain types of chemical reactions. One of the most important qualitative characteristics displayed by excitable media, and equally a common denominator between the diversity of phenomena mentioned

above, is the almost immediate damping out of signals below a certain threshold. On the other hand, signals exceeding this threshold propagate without damping.

The heart works by passing ionic current inside the muscle, thus triggering the rhythmic contractions that pump blood in and out. The ions move through small pores or *gates* in the cellular membrane, which can be either open (excitation state) or closed (rest state).

In nerve cells and cardiac cells the three abstract characteristics of excitable media are manifested as

- Rest cell membrane potential
- Threshold for opening the ionic gates in the cellular membrane
- The diffusive spreading of the electrical signals

The state of the membrane gates is random on a microscopic scale, but the probability of a given state can be modelled as a continuous function of the voltage, thus allowing an averaged macroscopic continuum description of the current flow.

The *FitzHugh-Nagumo* equations for excitable media describe the simplest physiological model with two variables, an *activator* and an *inhibitor*. In these heart models the activator variable corresponds to the electric potential, and the inhibitor is a variable that describes the voltage-dependent probability of the pores in the membrane being open and ready to transmit ionic current.

CHAOTIC DYNAMICS AND THE COMPLEX LANDAU-GINZBURG EQUATIONS

The *complex Landau-Ginzburg* equations provide a relatively simple way of modeling some aspects of the transition, displayed by many dynamical systems under the influence of strong external stimulus, from periodic oscillatory behavior into a chaotic state with gradually increasing amplitude of oscillations and decreasing periodicity.

Although their first use was to describe the theory of superconductivity, the complex Landau-Ginzburg equations are also generic in their nature (as are the FitzHugh-Nagumo equations), and examples of dynamical systems which you can model successfully using these equations are:

- The formation of vortices behind a slender obstacle in transversal fluid flow
- · Oscillating chemical reactions of the Belousov-Zhabotinsky type

In this model the complex Landau-Ginzburg equations simulate the dynamics of the spiral waves in excitable media.

Model Definition

The geometry here is a simplified 3D model of a heart with two chambers, represented with semispherical cavities¹.



Figure 5-1: Model geometry.

THE FITZHUGH-NAGUMO EQUATIONS

The equations are the following:

$$\begin{split} &\frac{\partial u_1}{\partial t} = \Delta u + (\alpha - u_1)(u_1 - 1)u_1 + (-u_2) \\ &\frac{\partial u_2}{\partial t} = \varepsilon(\beta u_1 - \gamma u_2 - \delta) \end{split}$$

Here u_1 is an *action potential* (the activator variable), and u_2 is a *gate variable* (the inhibitor variable). The parameter α represents the threshold for excitation, ε represents the excitability, and β , γ , and δ are parameters that affect the rest state and dynamics of the system.

The boundary conditions for u_1 are insulating, using the assumption that no current is flowing into or out of the heart. The initial condition defines an initial potential distribution u_1 where one quadrant of the heart is at a constant, elevated potential V_0 , while the rest remains at zero. The adjacent quadrant has instead an elevated value v_0 for the inhibitor u_2 . It is convenient to implement this initial distribution using the following logical expressions, where TRUE evaluates to 1 and FALSE to 0:

^{1.} Note that it is possible to import more realistic human/animal heart geometries, with anisotropies and inhomogenities as well as proper dimensions, into COMSOL Multiphysics using the CAD import capabilities.

$$\begin{split} &u_1(0,x,y,z) \,=\, V_0((x+d)>0)\cdot ((z+d)>0) \\ &u_2(0,x,y,z) \,=\, v_2((-x+d)>0)\cdot ((z+d)>0) \end{split}$$

Here d is equal to 10^{-5} , and it is added into the expressions to shift the elevated potential slightly off the main axes.

THE COMPLEX LANDAU-GINZBURG EQUATIONS

The complex Landau-Ginzburg equations are:

$$\begin{aligned} &\frac{\partial u_1}{\partial t} - \Delta(u_1 - c_1 u_2) = u_1 - (u_1 - c_3 u_2)(u_1^2 + u_2^2) \\ &\frac{\partial u_2}{\partial t} - \Delta(c_1 u_1 + u_2) = u_2 - (c_3 u_1 + u_2)(u_1^2 + u_2^2) \end{aligned}$$

The two variables u_1 and u_2 are the activator and inhibitor, respectively. The constants c_1 and c_2 are parameters reflecting the properties of the material. These constants also determine the existence and nature of the stable solutions.

As in the previous model, the boundary conditions are kept insulating. The initial condition, which gives a smooth transition step near z = 0, are the following:

$$u_1(0, x, y, z) = \tanh(z)$$
$$u_2(0, x, y, z) = -\tanh(z)$$

Modeling in COMSOL Multiphysics

The simplified geometry is quite straightforward to create using the drawing tools in COMSOL Multiphysics. The FitzHugh-Nagumo and Landau-Ginzburg equations are also readily entered in one of the PDE-based application modes².

It is important to note that these equations are strongly nonlinear. It is therefore necessary (especially in full 3D models like these) to use a much finer mesh or use higher element order than in these examples to get results with some degree of reliability for the time intervals of interest. This is particularly important in solving the complex Landau-Ginzburg equations, which describe inherently chaotic phenomena. They are highly sensitive to perturbations in the initial value and similarly to numerical errors during the course of the time-dependent solution. We recommend the use of the fourth-order Hermite element for the complex Landau-Ginzburg equation.

^{2.} With this stage completed, it would also be straightforward to replace the rather simple FitzHugh-Nagumo equations with a physiologically more realistic mathematical model.

For the reasons above, the results presented here are only intended as a first rough estimate of the qualitative behavior that you can expect the system to show under a given stimulus. Consequently, higher-order elements, finer meshing, and smaller relative and absolute time dependent tolerances clearly give quantitatively more correct simulation results. These improvements may require several hours of computational time to solve the equations, while the rough model described here should solve within around 20 minutes on a standard PC. When attempting these types of large models we strongly recommend the use of 64-bit platforms.

Results

THE FITZHUGH-NAGUMO EQUATIONS

The plots in Figure 5-2 below show the action potential u_1 . To visualize the solution on the inside, a quarter of the outside shell of the heart and one of the chamber surfaces are suppressed in the plot.

The parameters used in the model along with the initial pulse lead to a reentrant wave which travels around the tissue without damping in a characteristic spiral pattern.



Figure 5-2: Solution to the FitzHugh-Nagumo equations at times t = 125 s (left figure) and t = 500 s (right figure).

THE LANDAU-GINZBURG EQUATIONS

Figure 5-3 below shows the species u_1 at different times.



Figure 5-3: Solution to the Complex Landau-Ginzburg equations at times t=50 s (left) and t=75 s (right).

The equation parameters and initial condition used here lead the diffusing species (u_1) to display characteristic spiral patterns with growing complexity over time.

Modeling Using the Graphical User Interface

The instructions below describe how to create the model containing the FitzHugh-Nagumo equations for excitable media. To create the model containing the complex Landau-Ginzburg equations, follow the instructions from the beginning until you have completed the "Geometry Modeling" section. Then jump to the section called "Modifications."

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/heart_fhn

MODEL NAVIGATOR

- I Open COMSOL Multiphysics.
- 2 Select 3D from the Space dimension list.
- **3** Expand the **COMSOL Multiphysics** folder, then expand the **PDE Modes** subfolder, and finally select the **PDE**, **General Form** application mode.
- 4 Enter the dependent variables u_1 and u_2 by typing u1 u2 in the **Dependent variables** edit field at the bottom of the window.

5 Click OK to confirm and close the Model Navigator.

GEOMETRY MODELING

- I Click the **Sphere** button on the Draw toolbar on the left. Enter a radius of **54** for the sphere, then click **OK**.
- **2** Click the **Ellipsoid** button on the Draw toolbar. Enter 54, 54, and 70 for the x-, y-, and z-semiaxes, respectively. Click **OK**.
- **3** To create the egg-shaped solid, we will fuse the top half of the sphere with the bottom half of the ellipsoid. Creating the halves which we intend to fuse together, we need to divide the two solids across the middle. A convenient way of doing this is by using a work plane. To create a work plane select **Draw>Work-Plane Settings** and click on the **Vertices** tab. Expand the **ELP1** folder in the **Vertex Selection** on the left, select Vertex 1 and click the **>>** button to add it to the definition of the work plane. Then select Vertex 2, click **>>**, and finally select Vertex 6 and click **OK**.
- 4 In the 2D Geom2 geometry, click Zoom Extents in the Main toolbar at the top of the main window and draw a rectangle which completely covers the blue circle by using the Rectangle button on the left hand toolbar. (Note that the blue curves show the contours of the 3D geometry in the 2D plane.) Then select Draw>Embed and click OK.

The rectangular surface should now have been added to the 3D geometry, splitting both the sphere and the ellipse in two halves.

- 5 To create the separated geometry objects we need for the halves, select the rectangular surface, confirm the selection by right clicking, and then similarly select and confirm the spherical solid. With both the spherical solid and the rectangular surface selected, click on the **Coerce to Solid** button on the left toolbar. Then click the **Split object** button. Finally, remove the bottom half of the sphere by clicking on it to select it and press the Delete button on the keyboard.
- **6** Go back to the 2D geometry and select the rectangle, then choose **Draw>Embed**, and then click **OK**. Repeat Step 5 for the elliptical solid instead of the sphere. Note that this time you need to remove the top half of the elliptic solid (for the sphere it was the bottom half).
- 7 Select both objects by pressing Ctrl+A and go to Draw>Create Composite Object. Make sure that the Keep interior boundaries check box is cleared and click the Union button. Click OK.
- 8 To create the cavity inside the heart, press Ctrl+C and then Ctrl+V to create a copy of the egg-shaped solid (leave the **x**, **y**, and **z displacements** at zero). Then select **Scale**

from the **Draw>Modify** menu and set 2/3 as the scaling factor in the *x*, *y*, and *z* directions. Click **OK**.

- 9 Open the Create Composite Object dialog box again (Draw>Create Composite Object) and enter the expression C01-C02 in the Set formula edit field to subtract the smaller solid from the larger one. To see the cavities inside the solid you can click the Increase Transparency button on the left toolbar a couple of times.
- 10 To create the wall separating the two chambers click the Cylinder button on the Draw toolbar and enter 45 for the radius and 10 for the height. Type the values -5, 0, and -5 respectively in the x, y, and z edit fields in the Axis base point frame. Enter 1, 0, and 0 as the x, y, and z values in the Axis direction vector frame. Click OK.
- II Once again, open the **Create Composite Object** dialog box from the **Draw** menu. Then select both geometry objects, make sure the **Keep interior boundaries** check box is cleared and click the **Union** button. Click **OK**.

The final geometry should look as shown in Figure 5-1 on page 102.

VARIABLES AND EXPRESSIONS

Enter the following constants under **Options>Constants**:

NAME	VALUE
epsilon	0.01
gamma	1
beta	0.5
а	0.1
delta	0
V0	1
v0	0.3
d	1e-5

PHYSICS SETTINGS

Subdomain Settings

I Select **Physics>Subdomain Settings**. Enter the subdomain settings according to the following table:

SETTINGS	SUBDOMAIN I
F (row I)	(a-u1)*(u1-1)*u1-u2
F (row2)	epsilon*(beta*u1-gamma*u2-delta)

SETTINGS	SUBDOMAIN I
Γ (row I)	-u1x -u1y -u1z
Γ (row2)	0 0 0

2 Click the **Init** tab and enter the following initial conditions:

DEPENDENT VARIABLE	INITIAL CONDITION EXPRESSION
ul(t ₀)	V0*((x+d)>0)*((z+d)>0)
u2(t ₀)	v0*((-x+d)>0)*((z+d)>0)

- **3** If you can afford the increase in memory usage and solution time needed for better accuracy, click the **Element** tab and select **Hermite Quartic** from the **Predefined elements** list. Otherwise skip this step and use the default second-order Lagrange elements.
- 4 Click OK.

Boundary Settings

- I Select Physics>Boundary Settings.
- **2** Select all boundaries by pressing Ctrl+A, then enter boundary settings according to the following table:

SETTINGS	ALL BOUNDARIES
Туре	Neumann
g	0

3 Click OK.

COMPUTING THE SOLUTION

- I Choose Solve>Solver Parameters and select the Time dependent solver from the Solver list on the left side. In the Time stepping area on the General page, type 0:5:500 in the Times edit field.
- 2 In the Linear system solver list, select GMRES. Make sure that the Preconditioner is set to Incomplete LU, then click OK.
- **3** Click the **Solve** button on the Main toolbar (the **Solve** button is illustrated with an equals sign).

POSTPROCESSING AND VISUALIZATION

- I To create the right plot in Figure 5-2 on page 104, go to Postprocessing>Plot Parameters and select the Boundary check box on the General page. Clear all other check boxes in the Plot type area, then click OK.
- 2 Choose Options>Suppress Boundaries. Select Boundaries 2 and 4–9, then click OK to hide these boundaries.
- 3 Click the Postprocessing Mode button on the Main toolbar to see the plot.
- 4 To create the left plot in Figure 5-2 on page 104, go to Postprocessing>Plot Parameters and select 125 from the Solution at time list on the General page.

MODIFICATIONS FOR USING COMPLEX LANDAU-GINZBURG EQUATIONS

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/heart_clg

Follow the steps under the Model Navigator and Geometry Modeling sections just described. Then do the following:

Variables and Expressions

Enter the following constants in the **Options>Constants** list:

NAME	VALUE
c1	2
c3	-0.2

Subdomain Settings

I Go to **Physics>Subdomain Settings** and enter the following expressions for the PDE coefficient Γ:

DEPENDENT VARIABLE	$\Gamma_{\mathbf{x}}$	Γ _y	$\Gamma_{\mathbf{z}}$
ul	-u1x+c1*u2x	-u1y+c1*u2y	-u1z+c1*u2z
u2	-c1*u1x-u2x	-c1*u1y-u2y	-c1*u1z-u2z

2 Enter the following for the source term **F**:

DEPENDENT VARIABLE	F
ul	u1-(u1-c3*u2)*(u1^2+u2^2)
u2	u2-(c3*u1+u2)*(u1^2+u2^2)

3 Click the **Init** tab, then enter the following initial values:

DEPENDENT VARIABLE	INITIAL VALUE
ul(t ₀)	tanh(z)
u2(t ₀)	-tanh(z)

- **4** Follow the instructions under Boundary Settings above (the boundary conditions are the same for both equations).
- **5** Follow the steps under Computing the Solution with the following modifications:
 - Enter the times 0:5:75 in the Times edit field
 - Enter 1e-7 as the absolute and relative tolerances
- 6 Follow the steps under "Postprocessing and Visualization" on page 109 to produce the corresponding plots for the Landau-Ginzburg equations. Note that to produce the plot on the left in Figure 5-3 on page 105 you need to select 50 from the **Solution at time** list on the **General** page.

References

1. F.H. Fenton, E.M. Cherry, H.M. Hastings, and S.J. Evans, "Real-time computer simulations of excitable media: JAVA as a scientific language and as a wrapper for C and FORTRAN programs," *BioSystems 64*, pp. 73–96, 2002.

2. Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence*, Dover Publications, 2003.

3. J. Keener and J. Sneyd, Mathematical Physiology, Springer, 1998.

A Transport Problem

Background

This model features a pure stationary transport problems with a single convective term. It shows how to address oscillations and instabilities that arise from the numerical method. Here you stabilize a formulation with continuous Langrange elements by streamline diffusion and using a formulation with discontinuous elements with upwinding.

Model Definition

The following 1st-order equation models pure stationary transport in the direction of the constant vector β :

$$\begin{cases} \beta \cdot (\nabla u) = 0 \ \Omega \\ u = u_0 \qquad \Gamma \end{cases}$$

Here Γ is the inflow boundary, and the vector $\beta = (\beta_x, \beta_y)$ points into the domain Ω . The exact solution is constant on the characteristics

$$x = x_0 + t\beta_x$$
$$y = y_0 + t\beta_y$$

where *t* is a real number. Thus, if a characteristic intersects Γ at the point (x, y), then the value of *u* is $u_0(x, y)$ along the entire characteristic.

In this example, the domain Ω is the unit square and $\beta = (1, 1)$. Thus Γ corresponds to the left and lower edges of the square. Use the expression $u_0 = y > x$ to define the boundary conditions u = 1 on the left boundary and u = 0 on the lower boundary. The exact solution then equals

$$u(x,y) = \begin{cases} 1 & y > x \\ 0 & y < x \end{cases}$$

Streamline Diffusion

Streamline diffusion stabilizes oscillations and instabilities that arise from the numerical method. Streamline diffusion is an example of a *Petrov-Galerkin* method where the test-function space differs from the solution space. The modified test functions give rise to additional terms in the weak formulation of the problem. The streamline-diffusion contribution to the transport equation is

$$-\frac{h}{\sqrt{2}}(\hat{\beta_x u_x} + \hat{\beta_y u_y})(\hat{\beta_x u_x} + \hat{\beta_y u_y})$$

where the "hat" symbol denotes the corresponding test functions.

The Convection and Diffusion, Convection and Conduction, and Incompressible Navier-Stokes application modes have built-in artificial diffusion of several types, including streamline diffusion. This model explicitly adds the streamline-diffusion contributions.

The section "Stabilization Techniques" on page 433 in the *COMSOL Multiphysics Modeling Guide* describes general techniques for introducing contributions from artificial diffusion.

Upwinding Using Discontinuous Basis Functions

When you use continuous basis functions, the usual Galerkin method must be numerically stabilized, for example using streamline diffusion. Another approach is to use *discontinuous basis functions* (also know as the *discontinuous Galerkin method*).

The term $\beta \cdot (\nabla u)$ must be treated carefully because if u is discontinuous over an element boundary, this term is a δ -function on the boundary. Assume the vector field β is continuous within an element and has a continuous normal component across element boundaries (this is certainly true for the constant β used here). Define the *upstream* and *downstream* sides of an element boundary such that β points in the downstream direction, and define the normal vector n to be pointing in the opposite direction, from the *downside* to the *upside*. Also let the superscripts ^u and ^d indicate evaluation on the respective sides of a boundary.

It is then possible to assemble the contribution from the discontinuities as $\hat{u}^{d}(\beta \cdot n)(u^{d} - u^{u})$ (note that if β happens to be parallel to the boundary, "up" and "down" are undefined, but this is no problem because the factor $\beta \cdot n$ is then zero anyway). Using the test function on the downwind side gives sufficient stabilization

since it means that information is always taken from the upwind side and transported downwind (*upwinding stabilization*).

When using discontinuous elements, it does not make sense to impose a boundary condition on u by constraints because the value of u is not well defined on an element boundary. Instead use the same transport term, where you can imagine an element upstream of Γ with a value u_0 . Thus assemble a contribution $\hat{u}^d(\beta \cdot n)(u^d - u_0)$ at element boundaries belonging to the inflow boundary Γ .

Note: This upwinding method is not suited for problems with diffusion terms.

Results

The PDE is stationary but has a discontinuity in the solution. The model shows how you can prevent instabilities from propagating into areas where the solution should be constant by using streamline diffusion or upwinding.



The plot shows the discontinuities between elements stabilized with upwinding.

When setting up this model, think of the equation as a special case (c = 0) of a general convection-diffusion equation:

$$-\nabla \cdot (c\nabla u) + \beta \cdot \nabla u = f \Omega$$

Using the Convection-Diffusion application mode, this transport problem is easy to set up and solve. A second step adds explicit streamline diffusion.

Further steps change the model to use discontinuous elements and upwinding. In COMSOL Multiphysics all boundaries have an "up" and "down" side defined independent of the β direction. However, you can use the expression

```
(bx*dnx+by*dny<0)*(bx*dnx+by*dny)*test(down(u))*(down(u)-up(u))+(bx*unx+by*uny<0)*(bx*unx+by*uny)*test(up(u))*(up(u)-down(u))
```

for the interior mesh boundaries using an ultraweak term, and use the expression

```
(bx*dnx+by*dny<0)*(bx*dnx+by*dny)*test(down(u))*(down(u)-u0)+(bx*unx+by*uny<0)*(bx*unx+by*uny)*test(up(u))*(up(u)-u0)
```

for the inflow boundaries, which are independent of the definitions of "up" and "down."

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/transport_problem

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Open the Model Navigator, and select 2D from the Space dimension list.
- 2 In the list of application modes select

COMSOL Multiphysics>PDE Modes>PDE, Coefficient Form.

- 3 Make sure that Lagrange Quadratic elements are selected in the Element list.
- 4 Click OK.

GEOMETRY MODELING

I On the Draw menu point to Specify Objects, select Square, then click OK.

Because you do not change any settings in the **Square** dialog box, you create the default square with its corner at (0, 0) and with a side length of 1.

2 Press the **Zoom Extent** button on the Main toolbar.

OPTIONS AND SETTINGS

Constants and Expressions

- I From the **Options** menu select **Expressions>Global Expressions**.
- 2 Enter the following names and expressions; when done, click OK:

NAME	EXPRESSION
bx	1
by	1
u0	y>x

PHYSICS SETTINGS

Boundary Conditions

- I From the Physics menu choose Boundary Settings.
- 2 Enter the following settings in the **Boundary Settings** dialog box; when done, click **OK**.

SETTINGS	BOUNDARIES I, 2	BOUNDARIES 3, 4
Туре	Dirichlet	Neumann
r	u0	-

Subdomain Settings

- I From the Physics menu choose Subdomain Settings.
- **2** Select Subdomain 1.
- 3 Change these PDE coefficients in the Subdomain Settings dialog box:

PROPERTY	SUBDOMAIN I
c	0
f	0
d _a	0
β	bx by

- 4 Click the Weak tab.
- 5 Type -h*(bx*ux_test+bx*uy_test)/sqrt(2)*(bx*ux+bx*uy) in the weak edit field.
- 6 Click OK.

To add streamline diffusion automatically, instead use the Convection and Diffusion application mode. When in that mode open the **Subdomain Settings** dialog box, click the **Artificial Diffusion** button and select the **Streamline diffusion** check box to add stabilization by streamline diffusion.

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

Click the **3D Surface Plot** button on the Plot toolbar to visualize the solution.

Investigate the effect of streamline diffusion stabilization by turning it off and then setting the weak contribution to **0**. You can increase the accuracy by refining the mesh.

Continue with the same model using discontinuous elements and upwinding instead of Lagrange elements and streamline diffusions.

PHYSICS SETTINGS

Boundary Conditions

- I From the Physics menu choose Boundary Settings.
- 2 Select all the boundaries and click the Neumann boundary conditions button.
- 3 Click the Weak tab.
- 4 Enter the following settings in the **Boundary Settings** dialog box; when done, click **OK**.

TERM	ALL
weak	(bx*dnx+by*dny<0)*(bx*dnx+by*dny)*test(down(u))*(down(u)-u0)+ (bx*unx+by*uny<0)*(bx*unx+by*uny)*test(up(u))*(up(u)-u0)
dweak	0
constr	0

Subdomain Settings

- I From the Physics menu choose Subdomain Settings.
- 2 Select Subdomain 1, click the Weak tab, and enter these terms:

TERM	ALL
weak	0
dweak	0
constr	0
bnd.weak	(bx*dnx+by*dny<0)*(bx*dnx+by*dny)*test(down(u))*(down(u)-up(u))+(bx*unx+by*uny<0)*(bx*unx+by*uny)*test(up(u))*(up(u)-down(u))

3 Click OK.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

- I From the Postprocessing menu choose Plot Parameters.
- 2 On the Surface page clear the Smooth check box.
- 3 Click OK.

Discontinuities between elements are clearly visible in the plot.

Modeling Using COMSOL Script

I Create a square domain:

```
clear fem
fem.geom = square2;
fem.mesh = meshinit(fem);
```

2 Plot the domain to see the edge labels:

geomplot(fem,'edgelabel','on')
axis([-0.1 1.1 -0.1 1.1])

3 Specify the boundary conditions and the PDE:

fem.globalexpr = {'bx' 1 'by' 1 'u0' 'y>x'}; fem.bnd.h = {1 0 0 1}; fem.bnd.r = {'u0' 0 0 'u0'}; fem.equ.be = {{{'bx' 'by'}}};

4 Specify the streamline diffusion:

fem.equ.weak = '-h*(ux_test+uy_test)/sqrt(2)*(ux+uy)';

5 Select quadratic Lagrange elements and extend the mesh:

```
fem.shape = 2;
fem.xmesh = meshextend(fem);
```

6 Solve the system using streamline diffusion stabilization:

```
fem.sol = femstatic(fem);
postcont(fem,'u')
```

7 Now create the equations using the discontinuous Galerkin method. Set the boundary conditions:

8 Add the ultraweak contributions:

9 Solve the problem again:

```
fem.shape = 'shdisc(2,2,''u'')';
fem.xmesh=meshextend(fem);
fem.sol=femstatic(fem);
```

IO Plot without smoothing:

```
postsurf(fem, 'u', 'u', 'cont', 'off')
```

II You can also try to use discontinuous quad elements:

```
fem.mesh=meshmap(fem);
fem.xmesh=meshextend(fem);
fem.sol=femstatic(fem);
postsurf(fem,'u','u','cont','off')
```

12 An interesting experiment is to use the meshpoi command to get a regular triangular mesh:

```
fem.mesh=meshpoi(fem,10,10);
fem.xmesh=meshextend(fem);
fem.sol=femstatic(fem);
postsurf(fem,'u','u','cont','off')
```

Using this mesh, the numerical solution exactly represents the solution to the PDE problem.

An Integro-Partial Differential Equation³

Introduction

This example investigates how to solve the integro-partial differential equation

$$\frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) - \frac{4D_i}{D_o^2 - D_i^2} \varepsilon \sigma T^4 + \frac{4D_i}{D_o^2 - D_i^2} \varepsilon \sigma \int_0^L k(x, x') T(x')^4 \cdot \frac{dx'}{D_i} = \rho C_p \frac{\partial T}{\partial t}$$

where ρ is the density, C_p is the heat capacity, κ is the thermal conductivity, σ is Stefan's constant (the Stefan-Boltzmann constant), ϵ is the emissivity, and k(x, x') is the kernel corresponding to the radiation view factor. This equation arises in the physical description of 1D heat conduction and radiation along a pipe. Figure 5-4 shows the model geometry.

Before setting up the model, make the following assumptions:

- Inside the tube, neglect convection and consider only radiation and conduction.
- Assume blackbody radiation with $\varepsilon = 1$.
- Model heat transfer only in the *x* direction (assume θ symmetry).
- The pipe's outer wall is perfectly insulated so that no heat escapes to the outside world by either radiation or conduction.

The definition of the kernel k(x, x') is

$$1 - \frac{2\xi^3 + 3\xi}{2(\xi^2 + 1)^{3/2}}$$

where $\xi = |x - x'| / D_i$ as explained in Ref. 1.

^{3.} This model is courtesy of Daniel Smith and Ali Shajii of MKS Instruments, Wilmington, Mass., USA.



Figure 5-4: Model geometry and boundary conditions.

Also consider the following boundary conditions and initial condition:

$$T(0, t) = 300 + 1200 \tanh\left(\frac{t}{1 \text{ s}}\right) \text{ K}$$
$$T(L, t) = 300 \text{ K}$$
$$T(x, 0) = 300 \text{ K}$$

Modeling in COMSOL Multiphysics

To model the equation, use the Heat Transfer by Conduction application mode. In this application mode you can include the radiation effects in the source term, Q, using an integration-coupling variable.

To enter convolution integrals of the type needed here, use the dest operator, which forces COMSOL Multiphysics to evaluate the expression on which it operates on the destination points instead of the source points. In the expression k(x, x'), x' is the variable to integrate over, whereas the model does not integrate over x. To specify that x should remain a variable that can take on values from the entire domain, write it as dest(x) when defining the integration-coupling variable. In the case of a coupling variable, dest(f(x)) makes COMSOL Multiphysics evaluate f(x) on the destination domain and not the source domain.

Results





Figure 5-5: Temperature distribution along the pipe at t = 100 s.

It is easy to make a comparison with a radiation-free model by removing the source term from the heat transfer equation. The equation to solve now becomes

$$\frac{\partial}{\partial x} \left(\kappa \frac{\partial T}{\partial x} \right) = \rho C_p \frac{\partial T}{\partial t} \,.$$





Figure 5-6: The temperature distribution with radiation (dotted line) and without radiation (solid line).

Comparison with the Full 3D Radiation Model

To illustrate the validity of the 1D model, you can set up the entire stationary 3D model using the Heat Transfer Module. Its General Heat Transfer application mode handles surface-to-surface radiation boundary conditions, making it easy to verify the results (Figure 5-7).



Figure 5-7: 3D temperature distribution in the pipe.

Figure 5-8 compares the temperature distributions along the axial direction for the two models (1D and 3D). Clearly the results are in good agreement.



Figure 5-8: The temperature distributions from the 1D model (square markers) and the 3D model (triangle markers).

Reference

1. R. Siegel and J. Howell, *Thermal Radiation Heat Transfer*, 4th ed., Taylor & Francis Group, New York, 2001.

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/integro_partial

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select ID in the Space dimension list.
- **2** In the list of application modes select

COMSOL Multiphysics>Heat Transfer>Conduction>Transient analysis.

3 Click OK.

GEOMETRY MODELING

- Draw a line from -0.1 to 0.1. To do so, from the Draw menu select Specify Objects>Line, then enter the coordinates -0.1 0.1. Click OK.
- 2 Click the **Zoom Extents** button on the Main toolbar.

OPTIONS AND SETTINGS

Constants and Expressions

I From the **Options** menu select **Constants**. Enter the following names, expressions, and descriptions (the descriptions are optional); when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
kappa	13[W/(m*K)]	Thermal conductivity
rho	8700[kg/m^3]	Density
C_p	300[J/(kg*K)]	Heat capacity
sigma	5.67e-8[W/(m^2*k^4)]	Stefan's constant
epsilon	1	Emissivity
T_cold	300[K]	Temperature, cold end
DT_max	1200[K]	Maximum temperature difference
T_init	T_cold	Initial temperature
D_i	2.54[cm]	Inner diameter
D_o	D_i*1.1	Outer diameter

2 Go to the Options menu and select Expressions>Subdomain Expressions. Select Subdomain 1, then enter the following two expressions; when done, click OK.

NAME	EXPRESSION
Q_source	4*sigma/(D_o^2-D_i^2)*irrad
Q_loss	-4*D_i/(D_o^2-D_i^2)*sigma*epsilon*T^4

Coupling Variables

I From the **Options** menu select **Integration Coupling Variables>Subdomain Variables**, then enter the following variable (type the complete expression on a single line); when done, click **OK**.

NAME	EXPRESSION
irrad	T^4*(1-(2*(abs(x-dest(x))/D_i)^3+3*(abs(x-dest(x))/ D_i))/(2*(((x-dest(x))/D_i)^2+1)^1.5))

This is the integral term in the original equation, which you specify in this way using the dest operator. This coupling variable does not have a unit, so for other quantities that include the irrad variable (in this case, Q_source and the heat source setting that uses Q_source), COMSOL Multiphysics indicates inconsistent units.

PHYSICS SETTINGS

Subdomain Settings

I From the **Physics** menu select **Subdomain Settings**, then enter these expressions for the material properties:

NAME	EXPRESSION
$\boldsymbol{\delta}_{\text{ts}}$	1/60
κ	kappa
ρ	rho
C _p	С_р
Q	Q_source+Q_loss

The time-scaling factor δ_{ts} changes the time scale from seconds to minutes.

2 Click the Init tab. In the T(t₀) edit field type T_init, then click OK.

Boundary Conditions

I From the Physics menu select Boundary Settings.

- 2 In the Boundary selection list select Boundary 1. In the Boundary condition list select Temperature. In the T₀ edit field enter the expression T_cold+DT_max*tanh(t/ 1[s]). Dividing t by the time constant 1[s] makes the input to tanh unitless, as is appropriate.
- 3 Select Boundary 2, and in the Boundary condition list select Temperature. In the T₀ edit field type T_cold, then click OK.

MESH GENERATION

- I From the Mesh menu, select Initialize Mesh. Doing so results in 15 elements as you can see in the message log.
- 2 Select Mesh>Refine Mesh twice to end up with 60 elements.

COMPUTING THE SOLUTION

- I From the Solve menu select the Solver Manager.
- 2 On the Initial Value page, click the Initial value expression button. Click OK.
- 3 From the Solve menu select Solver Parameters.
- 4 On the General page go to the Solver list on the left side of the window and select Time dependent. In the Times edit field, type 0:1:100. Click OK.
- 5 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To reproduce the plot in Figure 5-5, follow these steps:

- I From the Postprocessing menu, select Domain Plot Parameters.
- 2 On the **General** page, go to the **Solutions to use** list and make sure that only time **100** is selected.
- 3 Click the Line/Extrusion tab, then click Apply.

Comparing the Results with a Radiation-Free Model

To compare the temperature distribution in the radiation model with that of a model without radiation, plot the temperature distributions from the two variations in the same figure.

I Again, go to Postprocessing>Domain Plot Parameters. Click the Line/Extrusion tab. Click Apply to plot the temperature *T*, then return to the General page and select the Keep current plot check box at the bottom of the window. Click OK.

- 2 From the Physics menu, select Subdomain Settings. Click the Physics tab. In the Q edit field type 0. Click OK. (This step removes the effect of the radiation on the temperature distribution.)
- **3** Click the **Solve** button.
- 4 Return to the Postprocessing>Domain Plot Parameters dialog box. Click the General tab. In the Solution to use list select 100. Click the Line/Extrusion tab. To change the line color, click the Line Settings button, then go to the Line color list and select Color. Click the Color button on the right and then choose your preferred color (for example, red). Click OK twice to return to the Domain Plot Parameters dialog box.
- 5 Click Apply. This step plots the temperature T in the same window as the previous temperature distribution, and the result should resemble Figure 5-6.

Visualizing the Effect of the Nonlocal Coupling

Note: This step requires that you run COMSOL Multiphysics with COMSOL Script or MATLAB.

To examine the effect of the nonlocal behavior, first export the model to the workspace (select **File>Export>FEM Structure** or press Ctrl+F), then enter these commands:

```
[K,L,M,N] = assemble(fem, 'T',100);
n = size(N,1);
spy([K,N',L;N,sparse(n,n),M]);
```

The pattern of nonzero entries reveals that the Jacobian for this problem is a full matrix.



The KdV Equation and Solitons

Introduction

The Korteweg-de Vries (KdV) equation, formulated in 1895 by Korteweg and de Vries, models water waves. It contrasts sharply to the Burgers equation, because it introduces no dissipation and the waves travel seemingly forever. In 1965, Zabusky and Kruskal named such waves as *solitons*.

The KdV equation with boundary conditions and initial value for this model is formulated as

$$u_t + u_{xxx} = 6uu_x$$
 in $\Omega = [-8, 8]$
 $u(-8, t) = u(8, t)$, periodic
 $u(x, 0) = -6 \operatorname{sech}^2(x)$

The equation models the steepening and dispersion of wavefronts but does not support a train of simple harmonic waves. Such trains comprise the wavecrests normally associated with the ocean: simply a momentary constructive interference of contributing waves moving at different speeds. However, the equation does support solitons, single "humps" that travel without changing shape or speed for unexpectedly long distances.

Indeed, Perry and Schimke (Ref. 2) concluded from shipboard oceanographic measurements that bands of choppy water in the Andaman Sea, which lies east of the Bay of Bengal and west of Burma and Thailand, are associated with large-amplitude oceanic internal waves. Satellite images have since clarified that these waves originate on shallow banks on a layer between warm and cool water. Further, Osborne and Burch (Ref. 1) analyzed oceanographic data in an effort to assess the forces of underwater current fluctuations associated with such waves on offshore drilling rigs. They concluded that the visually observed roughness bands are caused by internal solitons that follow the KdV equation (Ref. 3).

A more recent development is the application of the KdV equation to another type of waves—light waves. Today solitons have their primary practical application in optical fibers. Specifically, a fiber's linear dispersion properties level out a wave while the nonlinear properties give a focusing effect. The result is a very stable, long-lived pulse (Ref. 3). It is amazing that researchers have discovered a formula for such waves:

$$u = \frac{v}{\left[2\cosh^2\left(\frac{1}{2}\sqrt{v}\right)(x-vt-f)\right]}$$

This equation says that the pulse speed is what determines the pulse amplitude and the pulse width. The following simulation illustrates this effect. An initial pulse, which does not conform to the formula, immediately breaks down into two pulses of different amplitudes and speeds. The two new pulses do follow the formula and thus can travel forever. While the formula does not reveal how solitons interact, the simulation shows that they can collide and reappear, seemingly unchanged, just as linear waves do, another counterintuitive observation that is difficult to observe without predictions by computing.

Model Definition

In the model, the term uu_x describes the focusing of a wave and u_{xxx} refers to its dispersion. The balancing of these two terms permits waves to travel with their shape unchanged.

Because COMSOL Multiphysics does not evaluate third derivatives directly, you rewrite the original equation above as a system of two variables to solve it:

$$u_{1t} + u_{2x} = 6u_1u_{1x} u_{1xx} = u_2$$

Using the general form PDE, you need to define two dependent variables, u1 and u2, and identify the d_a , Γ , and F coefficients in the following equation:

$$d_a \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \Gamma = F$$

- Only the first equation has a time derivative, and it is with respect to u_1 , so only $d_a(1,1)$ is 1; the other three components are zero.
- The divergence in this model is a space derivative with respect to *x*. This means that the Γ component from the first equation is *u*₂, which you type as u2. The Γ component from the second equation is *u*_{1x}, which you express in COMSOL Multiphysics as u1x.
- The F term components are the right-hand side of the equations: F_1 is $6u_1u_{1x}$ (type 6*u1*u1x), and F_2 is u_2 (type u2).

The initial condition for u_1 uses a hyperbolic cosine function to provide an interesting wave form to start with. For u_2 , you must provide the second space derivative of this function to provide consistent initial conditions.

Results

The following plot shows how solitons collide and reappear with their shape intact.



References

1. A.R. Osborne and T.L. Burch, "Internal Solitons in the Andaman Sea," *Science*, vol. 208, no. 4443, pp. 451–460, 1980.

2. R.B. Perry and G.R. Schimke, "Large-Amplitude Internal Waves Observed off the Northwest Coast of Sumatra," *J. Geophys. Res.*, vol. 70, no. 10, pp. 2319–2324, 1965.

3. G. Strang, Applied Mathematics, Wellesley-Cambridge, 1986.
Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/kdv_equation

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select ID in the Space dimension list.
- **2** In the list of application modes, open the **COMSOL Multiphysics>PDE Modes** folder and then **PDE**, **General Form**. Select **Time-dependent analysis**.
- **3** Type u1 u2 (space separated) in the **Dependent variables** edit field to define two dependent variables, u_1 and u_2 .
- 4 Click OK.

OPTIONS AND SETTINGS

Set the *x*-axis range to [-9 9].

GEOMETRY MODELING

Draw a line from -8 to 8.

PHYSICS MODELING

Periodic Boundary Conditions

- I On the Physics menu, point to Periodic Conditions and then click Periodic Boundary Conditions.
- 2 On the Source page, select Boundary 1 and type u1 in the Expression column.
- 3 Press Enter. The constraint name pconstrl appears in the Constraint name column.
- **4** Click the **Destination** tab.
- 5 Select destination Boundary 2 and enter destination expression u1.
- 6 Click the Source Vertices tab.
- **7** Select Vertex 1, then click the **>>** button.
- 8 Click the Destination Vertices tab.
- **9** Select Vertex 2, then click the **>>** button.
- **IO** Click the **Source** tab.

II Select Boundary 1 and type u2 in the Expression column.

12 Press Enter. The constraint name pconstr2 appears in the Constraint name column.

I3 Click the **Destination** tab.

14 Select destination Boundary 2 and enter destination expression u2.

I5 Click the **Source Vertices** tab.

I6 Select Vertex 1 and click the **>>** button.

I7 Click the **Destination Vertices** tab.

I8 Select Vertex 2 and click the **>>** button.

I9 Click OK.

Boundary Mode

- I From the Physics menu, choose Boundary Settings.
- **2** Select both boundaries and click the **Neumann boundary condition** button. The default Dirichlet boundary condition would override the periodic boundary conditions.

SETTINGS	BOUNDARIES I, 2
Туре	Neumann
G(I)	0
G(2)	0

3 Click OK.

Subdomain Mode

I From the Physics menu, choose Subdomain Settings.

2 Select Subdomain 1, then enter the following PDE coefficients:

PROPERTY	VALUE
Γ(Ι)	u2
Г(2)	u1x
F(1)	6*u1*u1x
F(2)	u2
$d_a(11)$	1
$d_{a}(12)$	0
d _a (21)	0
d _a (22)	0

- **3** Click the **Init** tab.
- **4** Enter initial conditions.

PROPERTY	VALUE
u I (t ₀)	-6*sech(x)^2
u2(t ₀)	-24*sech(x)^2*tanh(x)^2
	+12*sech(x)^2*(1-tanh(x)^2)

5 Click OK.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box.
- 2 Type 0.1 in the Maximum element size edit field.
- 3 Click OK.
- 4 Click the Initialize Mesh toolbar button.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box.
- 2 Type linspace(0,2,81) in the **Times** edit field. This provides 81 equally-spaced time intervals from 0 to 2 seconds.
- **3** Click the **Time Stepping** tab.
- **4** Type **2** in the **Maximum BDF order** edit field in the **Advanced** area. This is to ensure stability in the time-stepping algorithm.

5 Click **OK** and then click the **Solve** button.



Figure 5-9: The solution to the KdV equation at 0.25 s (plot shows $-u_1$).

POSTPROCESSING AND VISUALIZATION

By default, the software plots the solution at the last time point. The model computes the negated solution, which is positive. The best way to visualize it is to extrude the result along time to create a time-series plot.

- I Open the Domain Plot Parameters dialog box.
- 2 Make sure all time steps are selected in the Solutions to use list on the General tab.
- 3 Click the Line/Extrusion tab.
- 4 Click the **Extrusion plot** button.
- **5** Select Subdomain 1 in **Subdomain selection** list.
- 6 Type the quantity -u1 in the Expression edit field in the y-axis data area.
- 7 Click OK. In the plot window that appears, click the Go to XY View button.

The Telegraph Equation

Introduction

This model examines how telegraph wire transmits a pulse of voltage using the *telegraph equation*. The telegraph equation models mixtures between diffusion and wave propagation by introducing a term that accounts for effects of finite velocity to a standard heat or mass transport equation.

This example models a small section of a telegraph wire and contains a study of the pulse of voltage moving along it. A parametric analysis provides results showing the shape of the pulse with varying damping coefficients.

Model Definition

The model is simple to define. The geometry is a one-dimensional line of length 1. To model the pulse, the initial condition is a bell-shaped voltage distribution. The boundary conditions define the flux at both ends of the wire section, which allows the voltage to vary freely.

DOMAIN EQUATIONS

The telegraph equation is the following:

$$u_{tt} + (\alpha + \beta)u_t + \alpha\beta u = c^2 u_{xx}$$

where:

- α and β are positive constants.
- *c* is the transport velocity.
- *u* is the voltage (the dependent variable).

The model begins with the values $\alpha = \beta = 0.25$ and c = 1.

BOUNDARY CONDITIONS

The boundary conditions at both ends are homogeneous Neumann conditions:

$$u_x(t,0) = 0$$
$$u_x(t,1) = 0$$

INITIAL CONDITION

The following equations for the initial condition describe a bell-shaped pulse with the highest point at 0.2 and a base width of 0.4:

$$u(0,x) = e^{-3\left(\frac{x}{0.2} - 1\right)^2}$$
$$u_t(0,x) = 0$$

Results

The figure below shows the results of the first simulation. It is clear that the pulse gets smoother as it propagates along the wire section. Figure 5-10 shows the shape of the pulse at t = 0, 0.5, and 1:



Figure 5-10: Shape of the pulse at $t = 0, 0.5, and 1: \alpha + \beta = 0.5$.

Small values of the term $\alpha\beta$ result in a smoother pulse compared to larger values, while the term $\alpha+\beta$ sets the amount of damping. The following plots shows the influence of



the term α + β on the damping. A value of α + β = 1 yields the pulse in Figure 5-11 at t = 0, 0.5, and 1:

Figure 5-11: Shape of the pulse at $t = 0, 0.5, and 1: \alpha + \beta = 1$.

In the figure above the height of the pulse decreases only slightly from the initial value. In Figure 5-12, the decrease in height is more pronounced owing to a damping term that is four times as large as the one used for Figure 5-11.



Figure 5-12: Shape of the pulse at $t = 0, 0.5, and 1: \alpha + \beta = 2$.

Applying the telegraph equation to 2D and 3D models follows the same protocol shown here but produces a more complex systems of equations.

Modeling in COMSOL Multiphysics

To set up the telegraph equation, use a Coefficient Form PDE application mode for time-dependent analysis.

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/telegraph_equation

MODEL NAVIGATOR

- I Select ID in the Space dimension list.
- 2 In the list of application modes, browse to COMSOL Multiphysics>PDE Modes>PDE, Coefficient Form.
- **3** Select **Time-dependent analysis, wave type**. Make sure that **Lagrange Quadratic** elements are selected in the **Element** list.
- 4 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- 2 Enter the following constants in the **Constants** dialog box:

NAME	EXPRESSION
С	1
alpha	0.25
beta	0.25

3 Click OK.

GEOMETRY MODELING

- I Click the Line button and draw a line of length 1 from 0 to 1 on the *x*-axis.
- 2 Click the **Zoom Extents** button on the Main toolbar.

PHYSICS SETTINGS

Boundary Conditions

- I From the Physics menu, choose Boundary Settings.
- 2 Select Boundaries 1 and 2. From the Boundary condition list select Neumann.
- 3 Click OK.

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

PROPERTY	SUBDOMAIN I
с	c*c
a	alpha*beta
f	-(alpha+beta)*ut
d_a	0
e_a	1

2 On the Coefficients page in the Subdomain Settings dialog box, enter these PDE coefficients (the e_a and d_a coefficients are the default values):

		(C)	<u>لک</u>
Equation			
$e_a \partial^2 u / \partial t^2 + d_a \partial u / \partial t + \nabla \cdot (-c_a \partial u / \partial t)$:⊽u - αu + γ) + au	+ β·∇u = f	
Subdomains Groups	Coefficients Ir	nit Element Weak	Color/Style
Subdomain selection PDE coefficients			
1 (default)	Coefficient	Value/Expression	Description
	c	c*c	Diffusion coefficient
	a	alpha*beta	Absorption coefficient
	F	-(alpha+beta)*ut	Source term
	e,	1	Mass coefficient
-	d	0	Damping/Mass coefficient
Group: default -	a	0	Conservative flux convection coeff.
	β	0	Convection coefficient
Select by group	Y	0	Conservative flux source term
Active in this domain			
		ОК	Cancel Apply Help

3 Click the **Init** tab and enter the following initial conditions:

PROPERTY	SUBDOMAIN I
$u(t_0)$	exp(-3*(x/0.2-1)^2)
$u_t(t_0)$	0

4 Click OK.

MESH GENERATION

Initialize the mesh and refine it once.

COMPUTING THE SOLUTION

- I From the Solve menu, choose Solver Parameters.
- 2 Click the Time Stepping tab.

- **3** Select the **Manual tuning of step size** check box. Type 0.002 in the **Maximum time step** edit field. To get a good solution to the telegraph equation with this mesh, a time step of around 0.002 is sufficient. It means that the solver takes about 500 steps.
- 4 Click OK.
- 5 Click the Solve button on the Main toolbar to run the analysis.

POSTPROCESSING AND VISUALIZATION

- I From the **Postprocessing** menu, choose **Plot Parameters**.
- **2** Select the **Keep current plot** check box and plot the solution at time 0, 0.5, and 1.
- 3 From the **Options** menu, choose **Constants**.
- 4 In the **Constants** dialog box, change alpha and beta to 0.5.
- 5 Click OK.
- **6** Click the **Solve** button.

Continue to investigate the effect of the α and β coefficients by changing their values.

Shallow Water Equations

The shallow water equations are frequently used for modeling both oceanographic and atmospheric fluid flow. Models of such systems lead to the prediction of areas eventually affected by pollution, coastal erosion, and polar ice-cap melting.

Comprehensive modeling of such phenomena using physical descriptions such as the Navier-Stokes equations can often be problematic, due to the scale of the modeling domains as well as through resolving free surfaces. The shallow water equations, of which there are a number of representations, provide an easier description of such phenomena.

This 1D model investigates the settling of a wave over a variable bed as a function of time. The initial wave and the shape of the bed are represented by mathematical relations so that it is easy to change parameters such as the wave amplitude or the bed's shape.

Introduction

This example uses the *Saint-Venant's* shallow water equations, which are the following:

$$\frac{\partial z}{\partial t} + \nabla \cdot (zv) = 0$$

and

$$\frac{\partial(zv)}{\partial t} + \nabla \cdot (zvv^{T}) + gz\nabla z_{s} - v\Delta(zv) = 0$$

where z is the thickness of the water layer (m), v is the velocity (m/s), g is the gravity constant (m/s²), and v is the kinematic viscosity (m²/s). The definition of the

thickness of the water layer, z, is z_s - z_f , where z_s and z_f are the measures in Figure 5-13 below. For further details, see Ref. 1.



Figure 5-13: Representative vertical section through the fluid domain showing the bed of a lake and the water surface.

Artificial Stabilization

With time, the flow develops discontinuities known as hydraulic jumps. Use artificial stabilization to replace the jumps by steep fronts that can be resolved on the grid. Small amplitude waves on still water of depth *z* move with velocity \sqrt{gz} . The maximal propagation velocity is $v_{phase} = |v| + \sqrt{gz}$ for water waves.

Stabilize the solution by adding artificial viscosity chosen to make the cell Reynolds number of order unity. To do so, add the term tune $v_{\text{phase}} \cdot h \cdot \frac{\partial}{\partial x}(vz)$ to the physical viscous momentum flux $v \cdot \frac{\partial}{\partial x}(zv)$. Here tune is a O(1) tuning parameter and h is the local element size. You add the contribution to the divergence term of the conservation law so that it does not affect the shock speeds. The modification is first order in element size.

Model Definition

This model studies a simple example of shallow water in a channel with bottom topography shown in Figure 5-14. Notice the difference in scale between the x and y directions.



Figure 5-14: Sea bed profile, z_{f} used in the model.

Dirichlet boundary conditions (v = 0) are implemented at both ends, while the physics are described by the equations above. The initial condition is a wave profile, which the following expression defines:

$$z_0 = 2 \cdot 10^{-2} - z_f + 5 \cdot 10^{-3} e^{\frac{-(x-3)^2}{1^2}}$$

where z_f is the analytical expression for the sea bed profile (see Figure 5-14). The elevation of the water surface is $z + z_f$, while Figure 5-15 shows $z_0 + z_f$.



Figure 5-15: The initial water surface profile, $z_0 + z_f$, and the sea bed profile, z_f .

Results and Discussion

The simulation runs for 60 seconds. Figure 5-16 shows the water surface and slope of the sea bed at six output times toward the beginning of the simulation.

Figure 5-16: The water level and the slope of the sea bed at six output times. Time spans from t = 0 to t = 15 at steps of 3 seconds.

The simulation clearly shows the influence of the topography of the sea bed on the elevation of the water surface. Another interesting visualization of the results is an animation, which is easy to create using COMSOL Multiphysics.

Modeling in COMSOL Multiphysics

The modeling procedure is straightforward using the PDE, General Form application mode with two dependent variables: Z and ZV. It is easy to define expressions, such as the one that describes the initial wave profile, z_0 , in the appropriate dialog box.

Reference

1. O. Pironneau, Finite Element Methods for Fluids, John Wiley & Sons, 1989.

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/shallow_water

Modeling Using the Graphical User Interface

- I Double-click the COMSOL Multiphysics icon to open the Model Navigator.
- 2 Select ID from the Space dimension list.
- 3 Select COMSOL Multiphysics>PDE Modes>PDE, General Form>Time-dependent analysis.
- 4 Type Z ZV in the Dependent variables edit field.
- 5 Click OK.

OPTIONS AND SETTINGS

- I Select Constants in the Options menu.
- **2** Enter the following constants:

EXPRESSION
1e-6
9.8
6
0.005
0.0015
0.1

3 Click OK.

4 Select Expressions>Scalar Expressions in the Options menu.

5 Enter the following scalar expressions:

NAME	EXPRESSION
Zf	a*exp(-(x-x0)^2)+k1*x
dZfdx	(-2*x+2*x0)*a*exp(-(x-x0)^2)+k1
Zs	Z+Zf
Z0	0.02-Zf+0.005*exp(-(x-3)^2/1^2)
vphase	abs(ZV/Z)+sqrt(ge*Z)
nu	nu1+vphase*h*tune

6 Click OK.

GEOMETRY MODELING

I Select Specify Objects>Line in the Draw menu.

- 2 Type 0 10 in the x edit field. This creates a line from 0 to 10 along the x-axis.
- 3 Click OK.
- 4 Click the **Zoom Extents** button in the main toolbar.

PHYSICS SETTINGS

Subdomain Settings

- I Select Subdomain Settings in the Physics menu.
- 2 Select Subdomain 1 in the Subdomain selection list.
- **3** Type ZV in the first edit field (first row) in the **Flux vector** area.
- **4** Type ZV*ZV/Z-nu*ZVx in the second edit field in the **Flux vector** area.
- 5 Click the F tab.
- 6 Type 0 in the first edit field in the Source term area.
- 7 Type -ge*Z*(Zx+dZfdx) in the second edit field in the Source term area.
- 8 Click the **Init** tab to set the initial conditions.
- **9** Type Z0 in the **Z(t₀)** edit field. Type 0 in all other edit fields.

IO Click OK.

Boundary Settings

- I Select Boundary Settings from the Physics menu.
- 2 Select both Boundary 1 and Boundary 2 from the Boundary selection list.
- **3** Click the **R** edit field.

- **4** Type **0** in the first edit field (first row).
- **5** Type -ZV in the second edit field.
- 6 Click OK.

MESH GENERATION

- I Select Free Mesh Parameters in the Mesh menu.
- 2 Type 0.05 in the Maximum element size edit field.
- **3** Click **Remesh** and **OK**.

COMPUTING THE SOLUTION

- I Select Solver Parameters in the Solve menu.
- 2 Type linspace(0,60,61) in the Times edit field in the Time stepping area.
- **3** Type 1e-5 in the **Relative tolerance** edit field.
- 4 Type 1e-7 in the Absolute tolerance edit field.
- 5 Click OK.
- **6** Click the **Solve** button.

POSTPROCESSING AND VISUALIZATION

The most interesting part of the results is obtained by looking at the scalar expression Zs, corresponding to the surface topography, at different times compared to the topography of the bottom.

- I Click the **Plot Parameters** button.
- **2** Click the **Line** tab.
- **3** Type Zs in the **Expression** edit field; then click **Apply**.
- 4 Click the **General** page.
- 5 Select 6 from the Solution at time list.
- 6 Select the Keep current plot check box.
- 7 Click the Line tab.
- 8 Type Zf in the **Expression** edit field.
- 9 Click OK.

Shell Diffusion

Introduction

A goal for many applications is to predict physics in thin structures, such as shells, without modeling the thickness of the structure. This is because large aspect ratios can cause meshing and geometry analysis problems. The model reported here demonstrates how to use the *tangential derivative variables* in COMSOL Multiphysics to solve partial differential equations in curved 3D shells and 2D boundaries without modeling their thickness.

Model Definition

The steel tank shown below has two pipe connections. One is grounded and the other connects to a dead current source. This model calculates the current density in the tank shell along with the potential distribution across the surface.



EQUATIONS

The fundamental equation to be solved is the current conduction, or charge conservation equation.

$$\nabla \cdot (-\sigma \nabla V) = 0 \tag{5-1}$$

Here, σ is the electric conductivity (S/m) and V is the electric potential (V).

The material is a 1 mm thick steel sheet with a conductivity of $4.032 \cdot 10^6$ S/m. You are working with a surface in 3D so there is no thickness in the model. To account for the charge conservation in Equation 5-1 you must multiply the current flux expression with the shell thickness *d*:

$$\nabla \cdot (-\sigma \, d\nabla V) = 0 \, .$$





Figure 5-17: Electric potential distribution across the surface (V).

Figure 5-17 reveals the potential distribution across the surface. Figure 5-18 shows a zoom of one of the pipe connections. The current arrows show clearly how the current is collecting toward the hole that is earthed. It is possible to see some current arrows on the opposite wall through the interior of the tank.



Figure 5-18: Arrow plot of the local current field.



Figure 5-19: Plot of the local magnitude of the electric current density (A/m^2) .

The plot of the magnitude of the local current density in Figure 5-19 is interesting because you can use it to calculate the resistive heating in the material as an extension to the model.

Modeling in COMSOL Multiphysics

The current conduction equation is modeled here using a Weak Form, Boundary application mode. For example, you can express the Poisson's equation $-\Delta u = F$ in 2D using the weak term sequence $-ux_test*ux-uy_test*uy+u_test*F$. Tangential derivative variables come into play only in the shell, which you represent as a boundary. To call the tangential derivative variables in COMSOL Multiphysics, add a T suffix to the variable name. For example, the tangential derivative corresponding to ux is uTx. To fully appreciate the weak-level formulation, some knowledge of the weak form in the finite element method is recommended. Further information is available in the section "Theoretical Background" on page 310 in the *COMSOL Multiphysics Modeling Guide*.

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/shell_diffusion

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator, select 3D in the Space dimension list. In the list of application modes, select COMSOL Multiphysics>PDE Modes>Weak Form, Boundary.
- **3** Type V in the **Dependent variables** edit field.
- 4 Click OK.

OPTIONS AND SETTINGS

I From the **Options** menu, choose **Constants**.

2 Enter these constants:

NAME	EXPRESSION
sigma	4.032e6
d	1e-3

3 Click OK.

GEOMETRY MODELING

Create the model by drawing the profile of revolution for the shell and then subtract a hole from the shell:

- I From the Draw menu, choose Work-Plane Settings.
- 2 Click the Quick tab and then click the y-z button. Click OK.
- **3** In the 2D drawing, double-click **SOLID** on the status bar to disable automatic coercion to solid objects.
- 4 Click the Line button on the Draw toolbar and draw an open polygon with corners in (0, 0), (0.15, 0), (0.15, 1), and (0.05, 1).
- 5 Click the Fillet/Chamfer button on the Draw toolbar.
- 6 Open the **BI** folder and select vertex 3 and 4 (the two corners in the polygon object). Make sure the **Fillet** button is selected and type 0.05 in the **Radius** edit field.
- 7 Click **OK** to create two fillets with radii of 0.05.
- 8 Select Revolve from the Draw menu and use the default settings.
- **9** Return to the 2D sketch and draw a centered circle around (0, 0.2) with a radius of 0.05.
- 10 Select Extrude from the Draw menu, type 0.2 in the Distance edit field, and click OK.
- II Press Ctrl+A to select both objects and then click the **Coerce to Face** button on the Draw toolbar.
- 12 Double-click on the geometry object with the right mouse button to open up the **Object Properties** dialog box. Select faces 16, 17, 19, 20, and 25–30 in the **Face** selection list and then click **Delete** to remove them.

BOUNDARY CONDITIONS

- I From the Physics menu, choose Boundary Settings.
- 2 Press Ctrl+A to select all boundaries.

- 3 On the Weak page, type sigma*d*(-VTx_test*VTy-VTy_test*VTy-VTz_test*VTz) in the weak edit field.
- 4 Click OK.

EDGE SETTINGS

- I From the Physics menu, choose Edge Settings.
- 2 Select edges 14, 15, 25, and 29, and type 400-V in the constr edit field.
- **3** Select edges 40–43, and type V in the **constr** edit field.
- 4 Click OK.

COMPUTING THE SOLUTION

Click the **Solve** button.

POSTPROCESSING AND VISUALIZATION

- I To generate Figure 5-17 on page 151, click the **Plot Parameters** button and select the **Boundary** plot type. Click the **Boundary** tab and select to plot V.
- 2 To generate Figure 5-18 on page 152, add an arrow plot. Click the Arrow tab and then select Plot arrows on: boundaries, Arrow type: Cone, Arrow length: normalized, Scale factor: 0.7. Let the x,- y-, and z-components of the arrow be -sigma*VTx, -sigma*VTy, and -sigma*VTz, respectively. Click OK. Click Decrease Transparency in the Plot toolbar several times to reset the transparency.
- **3** To generate Figure 5-19 on page 152, use a boundary plot again, but plot the expression sigma*sqrt(VTx^2+VTy^2+VTz^2).

Spherically Symmetric Transport

Introduction

Many models of industrial-transport problems allow the assumption that the problem is spherically symmetric. This assumption is of great importance because it eliminates two space coordinates and leaves a 1D problem that is computationally fast and has very small memory requirements. Some applications where spherical symmetry assumptions are useful include:

- · Reaction and diffusion in catalytic pellets in chemical reactors
- · Heat and mass transfer in the processing of upgraded iron-ore pellets
- · Any other process that takes place in beads that are nearly spherical

For spherical symmetry to be valid, the following assumptions must apply:

- The computational domain has a spherical shape
- The outer-perimeter boundary condition does not change with the position on the surface, that is, it does not vary with the space angles θ and ϕ
- At any given time for a time-dependent problem, the material properties depend only on the radial distance from the center, *r*, and not on the space angles θ and φ
- For a time-dependent problem, the initial condition depends only on the radial distance from the center, *r*, and not on the space angles θ and φ

Model Definition

This following example simulates the initial transient heating process of a pelletized piece of magnetite ore. This is the first step in the process of making hematite ore pellets, an important raw material for the steel industry.

During the initial heating of a magnetite pellet the temperature is in a range that allows you to disregard any phase change of moisture. Thus it is possible to use a transient heat-conduction equation with constant properties in spherical symmetry. You can also scale the equation for easy parameterization of the radius.

Figure 5-20 depicts some pellets together with a push pin as a scale reference.



Figure 5-20: Hematite pellets after drying and oxidation (end product)

The figure shows that these pellets are indeed not perfectly spherical. Nonetheless, this model takes advantage of the assumption of spherical symmetry.

DOMAIN EQUATIONS

Starting with the time-dependent heat conduction equation

$$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T) = Q$$

and expanding it in spherical polar coordinates, the result is the equation

$$\rho c_p \frac{\partial T}{\partial t} - k \left[\frac{1}{r^2 \partial r} \left(r^2 \frac{\partial T}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial T}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 T}{\partial \phi^2} \right] = Q$$

where ρ is the density (kg/m³), c_p gives the heat capacity (J/kg·K), k denotes the thermal conductivity (W/m·K), and Q is an internal heat source (W/m³). Further, r, θ , and φ are the space coordinates.

Assuming a perfect sphere of radius R_p and no change in temperature with differing space angles, or $\partial T/\partial \theta = \partial T/\partial \phi = 0$, gives

$$\rho c_p \frac{\partial T}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} \left(-kr^2 \frac{\partial T}{\partial r} \right) = Q$$

To avoid division by zero at r = 0, which causes numerical problems, multiply this equation by r^2

$$r^{2}\rho c_{p}\frac{\partial T}{\partial t} + \frac{\partial}{\partial r}\left(-kr^{2}\frac{\partial T}{\partial r}\right) = r^{2}Q.$$
(5-2)

Using a dimensionless radial coordinate \hat{r} by scaling the equation provides the option to quickly change the pellet's radius without changing or parameterizing the geometry size⁴. Introducing the dimensionless coordinate

$$\hat{r} = \frac{r}{R_p}, \quad \frac{\partial}{\partial r} = \frac{1}{R_p \partial r}$$

and substituting in Equation 5-2 leads to

$$\hat{r}^{2}\rho c_{p}\frac{\partial T}{\partial t} + \frac{\partial}{\partial \hat{r}}\left(\frac{-k\hat{r}^{2}}{R_{p}^{2}}\frac{\partial T}{\partial \hat{r}}\right) = \hat{r}^{2}Q$$
(5-3)

on the following domain:

$$r = 0$$
 $r = 1$

In a similar manner, it is possible to derive equations similar to Equation 5-3 for porous media flow, diffusion-reaction problems, and so on.

SYMBOL	NAME	VALUE
ρ	Density	2000 kg/m ³
c_p	Heat capacity	300 J/(kg·K)
k	Conductivity	0.5 W/(m·K)
$R_{\rm p}$	Pellet radius	0.005 m
Q	Heat source	0 W/m ³

The model uses the following material data:

^{4.} Note that scaling the variables to get well-conditioned problems is not necessary in COMSOL Multiphysics because the solvers use automatic variable scaling.

BOUNDARY CONDITIONS AND INITIAL CONDITIONS

Because of symmetry about r = 0, there is zero flux through this point, meaning $\frac{\partial T}{\partial \hat{r}} = 0$.

At the surface, r = 1, you use a convective heating expression with a heat transfer coefficient, h_s (W/(m²K), for the influx of heat (W/m²):

$$q_{\rm in} = h_s (T_{\rm ext} - T)$$
. (5-4)

This expression describes a hot gas with a temperature $T_{\rm ext}$ flowing around the pellet. $T_{\rm ext}$ is chosen at 95 °C. The heat transfer coefficient is set to 1000 W/(m²·K). The initial condition is set to 25 °C.



Figure 5-21: Temperature profiles from t = 0 to t = 10 s.

Figure 5-21 shows the temperature profiles from 0 to 10 seconds. Each line represents an increment of 0.5 s from the preceding line. From the topmost line it is clear that

D (

the center of the pellet has not reached steady state at 10 s. You can also plot the time evolution of the temperature at the center of the pellet.



Figure 5-22: Time evolution of temperature in the center of a pellet with radius $R_p = 5 \text{ mm}$

Figure 5-22 shows even more clearly how long the process must yet go before it reaches steady state. An interesting next step is to experiment with different particle radii, R_p , and different heating times.



Figure 5-23: Time evolution in the center of a pellet with radius $R_p = 2.5 \text{ mm}$

Simply reducing the radius somewhat lets the model reach steady state within 7 s.

Modeling in COMSOL Multiphysics

To implement Equation 5-3 and the boundary conditions of this problem, use the 1D time-dependent version of the PDE, General Form application mode:

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

For more information about using this application mode, please refer to the section "The Scalar General Form Equation" on page 250 in the *COMSOL Multiphysics Modeling Guide*.

The space coordinate in the model is \hat{r} . For typographical reasons we use rh in this model for "*r*-hat." Identifying the general form with Equation 5-3, the following settings generate the correct equation:

COEFFICIENT	EXPRESSION
ea	0
d_a	$\hat{r}^2 ho c_p$
Γ (flux vector)	$rac{-k\hat{r}^2}{R_p^2}rac{\partial T}{\partial \hat{r}}$
F (source term)	0

You must take special care when setting the heat-influx boundary condition on the pellet surface. $h_s(T_{\text{ext}} - T) = k \frac{\partial T}{\partial r}$ on the surface, so you need to compensate G accordingly:

$$G = \frac{\hat{r}^2}{R_p} h_s (T_{\text{ext}} - T)$$

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/spherical_symmetry_ore

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Open the Model Navigator. Click the Multiphysics button and click the Add Geometry button.
- 2 The Add Geometry dialog box appears. In the Space dimension list select ID, and in the Independent variables edit field enter rh theta phi. In the Unit system list select none. Click OK.
- 3 In the list of application modes, select

COMSOL Multiphysics>PDE Modes>PDE, General Form>Time-dependent analysis. Do *not* click OK yet.

- 4 In the Dependent variables edit field type T.
- **5** In the **Application mode name** edit field type **ore_pellet**.
- 6 In the Element list verify that Lagrange Quadratic is selected.
- 7 Click Add, then click OK.

OPTIONS AND SETTINGS

From the **Options** menu select **Constants**. Enter the following names, expressions, and (optionally) descriptions; when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
rho	2000	Density (kg/m^3)
ср	300	Heat capacity (J/(kg*K))
k	0.5	Conductivity (W/(m*K))
Rp	0.005	Pellet radius (m)
Qs	0	Heat source (W/m^3)
hs	1000	Heat transfer coefficient (W/(m^2*K))
Text	368	External temperature (K)
Tinit	298	Initial value (K)

GEOMETRY MODELING

- I Select the menu item Draw>Specify Objects>Line.
- 2 In the **rh** edit field enter 0 1, then click **OK**.
- 3 Click the **Zoom Extents** button on the main toolbar.

PHYSICS SETTINGS

Subdomain Settings

- I From the Physics menu choose Subdomain Settings.
- **2** Go to the **Subdomain selection** list and choose Subdomain 1. Enter the coefficients from the following table:

COEFFICIENT	VALUE/EXPRESSION	
Г	-k*rh^2/Rp^2*Trh	
F	rh^2*Qs	
e _a	0	
d _a	rh^2*rho*cp	

Note: Trh is COMSOL Multiphysics syntax for $\partial T/\partial(rh)$ if *T* and rh are defined variables.

- 3 Click the Init tab, and in the T(t₀) edit field enter Tinit.
- 4 Click OK.

Boundary Conditions

- I From the Physics menu choose Boundary Settings.
- 2 In the **Boundary Settings** dialog box enter the following settings; when done, click **OK**.

SETTINGS	BOUNDARY I	BOUNDARY 2
Boundary condition type	Neumann	Neumann
G	0	rh^2/Rp*hs*(Text-T)

Note: Clicking the **Neumann boundary condition** button disables the **R** edit field and sets it to 0. You can also manually type 0 in the **R** field to get a Neumann condition.

MESH GENERATION

- I Go to the Mesh menu and select Free Mesh Parameters.
- **2** Set the Maximum element size scaling factor to 0.4 and click Remesh.
- 3 Click OK.

COMPUTING THE SOLUTION

- I From the Solve menu choose Solver Parameters.
- 2 Find the Times edit field and enter the time steps as 0:0.25:10.
- 3 Click OK.
- 4 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To create a line plot with all temperature profiles for the time steps in Figure 5-21 on page 159, follow these steps:

- I Select the menu item Postprocessing>Domain Plot Parameters.
- **2** On the **General** page, in the **Plot type** area click the **Line/Extrusion plot** option button.
- **3** Verify that all the entries in the **Solutions to use** list are selected (the default).
- **4** Go to the Line/Extrusion page, and select the Line plot button.
- 5 In the y-axis data area go to the Predefined quantities list and select T.
- 6 Click the General tab. Click the Title/Axis button and enter the Title Temperature, the First axis label Dimensionless radius, and the Second axis label T (K).
- 7 Click OK.

To create a line plot with time evolution of the temperature at the center as in Figure 5-22 on page 160, follow these steps:

- I Select the menu item Postprocessing>Domain Plot Parameters.
- **2** Go to the **General** page, then to the **Plot type** area, and then select the **Point plot** option button.
- **3** Verify that all the entries in the **Solutions to use** list are selected (the default).

- 4 Click the **Point** tab.
- 5 In the y-axis data area find the Predefined quantities list and select T.
- 6 In the Boundary selection list choose 1.
- 7 Click the General tab. Click the Title/Axis button and set all fields to Auto.
- 8 Click **OK**, then **OK** again.

Finally, if you want to make an animation, execute the following instructions:

- I Click the **Plot Parameters** button on the Main toolbar (or press F12).
- 2 On the Animate page click the Start Animation button to launch the COMSOL Movie Player window, then click OK to close the Plot Parameters dialog box.

The Black-Scholes Equation

Introduction

There are different types of stock options:

- A *call option* is the right to buy a security at a specified price (called the exercise or strike price) during a specified period of time.
- A *put option* is the right to sell a security at a specified price during a specified period of time.

American options can be exercised at any time up to and including the day the option expires. European options can be exercised only on the day the option expires.

The famous Black-Scholes equation computes the cost u of a European stock option

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u}{\partial x^2} + rx \frac{\partial u}{\partial x} = ru$$

with the following parameters:

- *x*, the underlying asset price
- *r*, the continuous compounding rate of interest
- σ , the standard deviation of the asset's rate of return (also known as volatility)

A put option's value on the exercise day is

$$u(T,x) = \max(K-x,0),$$

where *K* is the strike price. The problem domain is infinite and consists of the entire real axis across the time domain $0 \le t \le T$.

The assumptions made in deriving the Black-Scholes equation are:

- The underlying stock pays no dividends.
- The price of the stock, one period ahead, has a log-normal distribution with mean and standard deviation that are constant over the life of the option.
- The existence of a risk-free interest rate which is constant over the life of the option.
- You can lend and borrow at the risk-free interest rate.
Black-Scholes derived an analytical expression for the solution to the above problem (available, for example, in the MATLAB Financial Toolbox). However, the formula works only for certain cases. For instance, you cannot use it when σ and r are functions of x and t. Using the PDE formulation, you can determine the price for such cases.

Model Definition

Because you must work within a finite domain $0 \le x \le X$, it becomes necessary to specify not only the boundary conditions for t = T but also for x = 0 and x = X. It is therefore necessary to analyze the problem's characteristics to determine the location of the input and output boundaries.

EQUATION DEFINITION

To put the equation in coefficient form, rewrite the equation as

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} \sigma^2 x^2 \frac{\partial u}{\partial x} \right) + \left(rx - \frac{\partial}{\partial x} \left(\frac{1}{2} \sigma^2 x^2 \right) \right) \frac{\partial u}{\partial x} - rc = 0.$$

In the following, denote

$$\bar{r} = rx - \frac{\partial}{\partial x} \left(\frac{1}{2}\sigma^2 x^2\right)$$

To reduce the problems with inflow boundaries, start by considering a put option: Study the value of a put option at a strike price K = 40 with $\sigma = 0.3$ and r = 0.12.

BOUNDARY CONDITIONS AND INITIAL CONDITIONS

Make the domain be $0 \le x \le 80$ with time running from 12 to 0. Then the initial condition at t = 12 and x = 80 is 0 based on the put option's value. The initial condition in the region $0 \le x \le 40$ varies linearly from 0 to 40. At the end of the simulation domain, the boundary is free (use a homogeneous boundary condition).

Modeling in COMSOL Multiphysics

Model the Black-Scholes equation using the following approach:

- Create a 1D time-dependent model, using the time-stepping algorithm to solve for c as a function of x and t, the time. The time steps go backward in time. Using a variable substitution to reverse the sign of the time, the d_a coefficient becomes -1.
- To model the initial condition, use the logical expression (x<40)*(40-x). This means that in the areas where x > 40, the initial value is zero.

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/black_scholes_put

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select ID in the Space dimension list.
- 2 In the application mode list, open COMSOL Multiphysics>PDE Modes and then PDE, Coefficient Form.
- 3 Select Time-dependent analysis. Make sure Lagrange Quadratic elements are selected.
- 4 Click OK.

OPTIONS AND SETTINGS

- I Open the Axes/Grid Settings dialog box.
- 2 In the x min and x max edit fields, type -1 and 81, respectively.
- 3 Click OK.
- **4** Open the **Constants** dialog box and enter the following constants:

NAME	EXPRESSION
r	0.12
sigma	0.3

5 Click OK.

GEOMETRY MODELING

Draw a line from 0 to 80:

- I On the Draw menu, point to Specify Objects and then click Line.
- **2** In the \mathbf{x} edit field, type 0 80.
- 3 Click OK.

PHYSICS MODELING

Boundary Conditions

I Open the Boundary Settings dialog box.

- **2** Select the left boundary and click the **Neumann boundary condition** button. Use the default zero Dirichlet condition on the right boundary.
- 3 Click OK.

Subdomain Settings

- I Open the Subdomain Settings dialog box.
- 2 Select Subdomain 1 and enter the PDE coefficients as follows:

COEFFICIENT	VALUE
c	1/2*sigma^2*x^2
a	r
f	0
d _a	- 1
e _a	0
β	(-r+sigma^2)*x

3 Click the lnit tab and type (x<40)*(40-x) in the $u(t_0)$ edit field. Click OK.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box.
- 2 On the Global page, type 2 in the Maximum element size edit field.
- 3 Click OK.
- 4 Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box.
- **2** Enter 12:-0.5:0 in the **Times** edit field to step backward in time from 12 to 0 with 0.5 decrement.
- 3 Click OK.
- 4 Click the **Solve** button.
- Modeling Using the Programming Language
- I Clear the FEM structure and set the problem dimension:

```
clear fem1
fem1.dim=1;
fem1.form='coefficient';
fem1.shape=2;
```

2 Specify the constants:

fem1.const={'r',0.12,'sigma',0.3};

3 Create the geometry:

fem1.geom=solid1([0 80]);

4 Specify the boundary conditions:

fem1.bnd.h={0 1};
fem1.bnd.r={0 0};

5 Specify the PDE coefficients.

```
fem1.equ.c={{{'1/2*sigma^2*x^2'}}};
fem1.equ.be={{{'(-r+sigma^2)*x'}};
fem1.equ.a='r';
fem1.equ.da=-1;
```

6 Specify initial condition:

fem1.equ.init={{'(x<40)*(40-x)'}};

7 Generate the mesh:

```
fem1.mesh=meshinit(fem1, 'hmax',2);
fem1.xmesh=meshextend(fem1);
```

8 Solve the problem and plot the results:

```
fem1.sol=femtime(fem1,'tlist',[12:-0.5:0],'report','on');
postcrossplot(fem1,1,1,'surfdata','u')
```

The Two-Term Boltzmann Equation⁵

Introduction

Fluid models of plasma dynamics require a full and consistent set of electron transport and rate coefficients. In the drift-diffusion approximation for the electron density, n, and the mean electron energy, $\bar{\varepsilon}$, the electron mobility, μ_e , electron diffusivity, D_e , and energy mobility, μ_{ε} , energy diffusivity, D_{ε} are required inputs, and are functions of background neutral particle number density, N, and mean electron energy. The rate coefficients, k_j , for the inelastic collisions occurring in a plasma are strong functions of the electron energy distribution function (EEDF). In general, the EEDF cannot be assumed to be Maxwellian; a better approximation is obtained by solving a two-term truncation of a spherical-harmonic expansion in velocity space of the Boltzmann equation (Ref. 1). The rate coefficients and transport properties are then computed as appropriate integrals over the EEDF. The inputs are the gas temperature, neutral particle number density, electric field, and a set of electron-impact cross sections. Each electron-impact cross section is a table of collision cross section versus energy.



Figure 5-24: Plasma discharge in an oxygen-filled glass sphere. Image courtesy of Technorama, Switzerland.

In this example, you solve the two-term Boltzmann equation for an oxygen plasma to obtain the EEDF. The electron-oxygen collisions occurring in an oxygen plasma are listed in Table 5-1 and the corresponding cross sections are plotted in Figure 5-25.

^{5.} This model is courtesy of Daniel Smith of MKS Instruments, Wilmington, Mass., USA.

(The model neglects electron-electron collisions, although their inclusion would be relatively straightforward; for further details on this issue, see Ref. 1.)



Figure 5-25: Electron-impact cross sections for molecular oxygen.

INDEX	LEGEND	REACTION	ТҮРЕ	$\Delta\epsilon$ (eV)
I	mom	$O_2 + e \rightarrow O_2 + e$	Momentum	0
2	att	$O_2 + e \rightarrow O + O^-$	Attachment	0
3	rot	$O_2 + e \rightarrow O_2(rot) + e$	Excitation	0.02
4	vibl	$O_2 + e \rightarrow O_2(v=1) + e$	Excitation	0.19
5	vib2	$O_2 + e \rightarrow O_2(v=1) + e$	Excitation	0.19
6	vib3	$O_2 + e \rightarrow O_2(v=2) + e$	Excitation	0.38
7	vib4	$O_2 + e \rightarrow O_2(v=2) + e$	Excitation	0.38
8	vib5	$O_2 + e \rightarrow O_2(v=3) + e$	Excitation	0.75
9	vib6	$O_2 + e \rightarrow O_2(v=3) + e$	Excitation	0.75
10	excl	$O_2 + e \rightarrow O_2(a^{\dagger}\Delta) + e$	Excitation	0.977
11	exc2	$O_2 + e \rightarrow O_2(b^1\Sigma) + e$	Excitation	1.627
12	exc3	$O_2 + e \rightarrow O + O + e$	Dissociation	4.5

TABLE 5-1: ELECTRON-OXYGEN COLLISIONS IN AN OXYGEN PLASMA.

INDEX	LEGEND	REACTION	ТҮРЕ	$\Delta\epsilon$ (eV)
13	exc4	$O_2 + e \rightarrow O + O + e$	Dissociation	6.0
14	exc5	$O_2 + e \rightarrow O + O + e$	Dissociation	8.4
15	exc6	$O_2 + e \rightarrow O + O + e$	Dissociation	9.97
16	ion l	$O_2 + e \rightarrow O_2^+ + 2e$	Ionization	12.06

TABLE 5-1: ELECTRON-OXYGEN COLLISIONS IN AN OXYGEN PLASMA.

Model Definition

The assumption of a Maxwellian EEDF can lead to substantial errors in the rate coefficients. This is particularly true in electronegative gases with a low ionization threshold where attachment removes the low-energy electrons and strong ionization "chops off" the tail of the EEDF. The two-term Boltzmann equation accounts for these effects. Here, a truncation of the velocity-space spherical-harmonic expansion

$$f(\mathbf{v}) = f_0(v) + f_1(v)\cos\theta + \dots, \quad \cos\theta \equiv \mathbf{v} \cdot \mathbf{E}/(vE)$$

inserted in the full Boltzmann equation results in a system of two PDEs: one for the isotropic part, f_0 , and one for the first anisotropic perturbation, f_1 , of the EEDF. These two first-order PDEs can be combined into a single second-order PDE for f_0 :

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} & \left(\frac{\varepsilon}{3\sigma_{\rm m}} \frac{\partial f_0}{\partial \varepsilon} \left(\frac{E}{N} \right)^2 + \sigma_{\varepsilon} \varepsilon^2 \left(f_0 + \frac{k_{\rm B} T}{e} \frac{\partial f_0}{\partial \varepsilon} \right) \right) = S, \qquad f_1 = \frac{E}{N} \frac{1}{\sigma_{\rm m}} \frac{\partial f_0}{\partial \varepsilon} \\ \sigma_{\varepsilon} &= \sum_{k \in \{\text{elastic}\}} 2 \frac{m_e}{M} \sigma_k \qquad S = \sum_{k \notin \{\text{elastic}\}} C_k \\ C_{k \in \{\text{attachment}\}} &= -\varepsilon \sigma_k(\varepsilon) f_0(\varepsilon) \\ C_{k \in \{\text{exc., dissoc.}\}} &= -[\varepsilon \sigma_k(\varepsilon) f_0(\varepsilon) - (\varepsilon + \Delta \varepsilon_k) \sigma_k(\varepsilon + \Delta \varepsilon_k) f_0(\varepsilon + \Delta \varepsilon_k)] \\ C_{k \in \{\text{ionization}\}} &= -[\varepsilon \sigma_k(\varepsilon) f_0(\varepsilon) - 2(2\varepsilon + \Delta \varepsilon_k) \sigma_k(2\varepsilon + \Delta \varepsilon_k) f_0(2\varepsilon + \Delta \varepsilon_k)] \end{aligned}$$
(5-5)

Here f_0 and f_1 are functions of $\varepsilon = m_e v^2 / (2e)$, the electron energy in electronvolts; σ_m is the effective momentum-transfer cross section (m²); m_e is the electron mass (kg); M is the mass of an oxygen molecule (kg); T is the neutral gas temperature (K); k_B is Boltzmann's constant, e is the elementary charge, E is the external electric field (V/m); N is the number density of molecular oxygen; and each σ_k (m²) is a cross section corresponding to collision k. Notice that the source term, S, for the inelastic collisions is nonlocal in energy space. The interpretation of this is that electrons with energy above the activation threshold, $\Delta \varepsilon$, inelastically collide with neutral O₂ molecules, which results in the electrons being reinserted into the EEDF at a lower energy. In the

case of ionization (k = 16), two equally energetic electrons are reinserted into the EEDF (the two electrons having equal energy is a simplifying assumption whose validity is best for low energies; see Ref. 1). For attachment, electrons are simply removed from the EEDF. Further, f_0 is subject to the integral constraint

$$\lambda \equiv \int_{0}^{\infty} \varepsilon^{1/2} f_0 d\varepsilon - 1 = 0$$
 (5-6)

and the boundary conditions

$$f_0 = 0 \qquad \text{at } \varepsilon = \infty$$

$$\frac{\partial f_0}{\partial \varepsilon} = 0 \qquad \text{at } \varepsilon = 0$$
(5-7)

Having solved for $f_0(\varepsilon)$, you compute the mobility and the diffusion coefficient as

$$\mu_e N(\bar{\varepsilon}) = -\frac{\gamma}{3} \int_{0}^{\infty} \frac{\varepsilon}{\sigma_m} \frac{\partial f_0}{\partial \varepsilon} d\varepsilon$$

$$D_e N(\bar{\varepsilon}) = \frac{\gamma}{3} \int_{0}^{\infty} \frac{\varepsilon}{\sigma_m} f_0 d\varepsilon$$
(5-8)

where $\gamma \equiv (2e/m_e)^{1/2}$ and $\bar{\epsilon}$ is the mean electron energy in electronvolts:

$$\bar{\varepsilon} = \int_{0}^{\infty} \varepsilon^{3/2} f_0(\varepsilon) d\varepsilon$$
 (5-9)

In the energy equation, the energy mobility and the energy-diffusion coefficient are

$$\begin{split} \mu_{\varepsilon} N(\bar{\varepsilon}) &= -\frac{\gamma}{3\bar{\varepsilon}} \int_{0}^{\infty} \frac{\varepsilon^{2}}{\sigma_{\rm m}} \frac{\partial f_{0}}{\partial \varepsilon} d\varepsilon \\ D_{\varepsilon} N(\bar{\varepsilon}) &= \frac{\gamma}{3\bar{\varepsilon}} \int_{0}^{\infty} \frac{\varepsilon^{2}}{\sigma_{\rm m}} f_{0} d\varepsilon \end{split} \tag{5-10}$$

The all-important rate coefficients are computed from the following integrals:

$$k_j = \gamma \int_0^\infty \sigma_j(\varepsilon) \varepsilon f_0(\varepsilon) d\varepsilon$$
 (5-11)

For comparative purposes, you can plot the computed rate coefficients against the ones calculated by assuming a Maxwellian EEDF

$$f_0 = \frac{2}{\sqrt{\pi}} \tau_e^{-3/2} \exp\left(-\frac{\varepsilon}{\tau_e}\right)$$

where the electron "temperature," τ_e (eV), is related to the mean electron energy by $\tau_e = 2\bar{\epsilon}/3$.

Results and Discussion

Figure 5-26 compares the Maxwellian EEDF to the EEDF computed from the twoterm Boltzmann equation. The strong ionization at high electron energies tends to "chop off" the high energy tail in the EEDF. In the case of oxygen, attachment is also strong and it tends to "mop up" the low-energy electrons.



Figure 5-26: EEDF for an oxygen plasma calculated using the Maxwellian approximation (black) and the two-term Boltzmann equation (red).

Figure 5-27 similarly compares the ionization rate coefficient, k_{16} , calculated with the EEDF from the two-term Boltzmann approximation with the corresponding Maxwellian result. As the figure shows, the Maxwellian EEDF overpredicts the rate coefficient for ionization—at a mean electron energy of 2 eV by nearly 10 orders of magnitude.

Furthermore, the calculated rate coefficients for all 16 electron-impact reactions are plotted in Figure 5-28, and Figure 5-29 displays the corresponding transport properties ($\mu_e N$, $D_e N$, $\mu_e N$, and $D_e N$) as functions of $\bar{\epsilon}$.



Figure 5-27: Ionization rate coefficient calculated for Maxwellian (black) and non-Maxwellian (red) EEDF.



Figure 5-28: Rate coefficients calculated using the two-term Boltzmann equation.



Figure 5-29: Transport properties calculated using the two-term Boltzmann equation.

These computed rate coefficients and transport properties can be fed into a full fluid model for the electron density and energy, either as a look up table or by approximating the rate coefficients with a curve fit.

Finally, using COMSOL Script or MATLAB, the effect of the inelastic collisions on the Jacobian matrix can be visualized using the spy command. The sparsity-pattern plot in Figure 5-30 shows that the Jacobian matrix is different from the traditional banded matrix formed with a regular 1D convection-diffusion equation. Because the inelastic collisions dominate the system, it is essential that they contribute to the Jacobian matrix.

Modeling in COMSOL Multiphysics

To solve Equation 5-5, use a PDE, General Form application mode in 1D, letting the *x*-coordinate represent the electron energy expressed in electronvolts. An additional auxiliary 2D geometry allows you to handle the nonlocal couplings in the source term, *S*, using extrusion coupling variables (see the section "Extrusion Coupling Variables" on page 261 of the *COMSOL Multiphysics User's Guide* for further details). To implement the integral constraint Equation 5-6, use an integration coupling variable and COMSOL Multiphysics' ODE interface; for descriptions of these features, see the

sections "Integration Coupling Variables" on page 255 of the *COMSOL Multiphysics User's Guide* and "Solving ODEs and Global Equations" on page 286 of the *COMSOL Multiphysics Modeling Guide*, respectively. The collision cross-section data, obtained from Ref. 2, is included in a set of text files that comes with your COMSOL Multiphysics installation.



Figure 5-30: Sparsity-pattern plot of the assembled Jacobian matrix. The far off-diagonal elements are a result of the nonlocal couplings.

References

1. G.J.M. Hagelaar and L.C. Pitchford, "Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models," *Plasma Sources Sci. Technol.*, vol. 14, pp. 722–733, 2005.

2. The Siglo Database, CPAT and Kinema Software, http://www.siglo-kinema.com/database/index.htm

Model Library path:

COMSOL_Multiphysics/Equation-Based_Models/two_term_boltzmann

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

I Start COMSOL Multiphysics.

- 2 From the Space dimension list in the Model Navigator, select ID.
- 3 From the Application Modes list, choose COMSOL Multiphysics>PDE Modes>PDE, General Form.
- 4 In the **Dependent variables** edit field, change the name to f.
- 5 Click Multiphysics, then click Add.
- 6 Click Add Geometry. From the Space dimension list, select 2D. Click OK.
- 7 Click OK to close the Model Navigator.

GEOMETRY MODELING

- I Click the **GeomI** tab.
- 2 Shift-click the Line button on the Draw toolbar. In the **Coordinates** area, enter 0 500 in the **x** edit field to draw a line between x = 0 and x = 500. Click **OK**.
- 3 In the same manner, draw a second line between x = 500 and x = 550.
- 4 Click the **Zoom Extents** button on the Main toolbar.
- **5** Click the **Geom2** tab.
- **6** Using the **Rectangle** dialog box, which you open by shift-clicking the **Rectangle**/ **Square** button on the Draw toolbar, draw two rectangles, R1 and R2, with the properties in the table below. Click **OK** after specifying each rectangle.

PROPERTY RI		R2
Width	250	250
Height	50	50
Base	Corner	Corner
x	0	250
у	0	0

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

OPTIONS AND SETTINGS

Model Settings

Because this *x*-coordinate in this model signifies energy, and the energy is expressed in the nonstandard unit electronvolt, disable COMSOL Multiphysics' unit support:

- I From the Physics menu, open the Model Settings dialog box.
- 2 On the Geom2 page, select None from the Base unit system list.
- 3 Click the Geom I tab, and do the same for this geometry.

4 Click **OK** to close the dialog box.

Constants

I From the **Options** menu, open the **Constants** dialog box.

2 Define the following constants (the descriptions are optional); when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION	
е	1.602e-19	The elementary charge (C)	
m_e	9.109e-31	Electron mass (kg)	
gam	<pre>sqrt(2*e/m_e)</pre>	Energy-velocity conversion factor	
k_B	1.381e-23	Boltzmann's constant (J/K)	
N_A	6.022e23	Avogadro's constant (1/mol)	
T_g	300	Gas temperature (K)	
tau_g	k_B*T_g/e	Gas temperature (eV)	
tau_ei	0.025	Initial electron temperature (eV)	
M_02	0.032/N_A	O2 molecule mass (kg)	
EoN	32	Reduced electric field (Td)	
dH3	0.02	Threshold energy, reaction 3 (eV)	
dH4	0.19	Threshold energy, reaction 4 (eV)	
dH5	0.19	Threshold energy, reaction 5 (eV)	
dH6	0.38	Threshold energy, reaction 6 (eV)	
dH7	0.38	Threshold energy, reaction 7 (eV)	
dH8	0.57	Threshold energy, reaction 8 (eV)	
dH9	0.75	Threshold energy, reaction 9 (eV)	
dH10	0.977	Threshold energy, reaction 10 (eV)	
dH11	1.627	Threshold energy, reaction 11 (eV)	
dH12	4.5	Threshold energy, reaction 12 (eV)	
dH13	6.0	Threshold energy, reaction 13 (eV)	
dH14	8.4	Threshold energy, reaction 14 (eV)	
dH15	09.97	Threshold energy, reaction 15 (eV)	
dH16	12.06	Threshold energy, reaction 16 (eV)	

Here, EoN is the value of the *reduced electric field*—that is, the ratio between the electric field and the neutral particle number density—expressed in *townsend* (1 Td = 10^{-21} V·m²). When computing the solution, you use EoN as the solver parameter.

Functions

Next define functions sig1, ..., sig16 that encode the electron-impact cross sections $\sigma_1, ..., \sigma_{16}$ for the collisions listed in Table 5-1 using data provided in text files.

- I From the **Options** menu, open the **Functions** dialog box.
- 2 Click New. In the Function name edit field, type sig1.
- 3 Click the Interpolation option button.
- 4 In the Use data from list, select File. Browse to the folder models/ Equation-Based_Models in your COMSOL Multiphysics installation directory. Select the file two_term_boltzmann_cs1.txt, then click OK.
- **5** From the **Interpolation method** list, select **Linear**. From the **Extrapolation method** list, select **Specific number**, then set the **Value outside range** to 0.
- 6 Repeat Steps 2–5 to define the functions sig2, ..., sig16 by loading, in turn, the text files two_term_boltzmann_cs2.txt, ..., two_term_boltzmann_cs16.txt.
- 7 Click **OK** to close the **Functions** dialog box.

Global Expressions

I Choose Options>Expressions>Global Expressions.

NAME	EXPRESSION	DESCRIPTION
sigma_eps	2*(m_e/M_02)*sig1(x)	Elastic-collision cross section
sigma_m	sig1(x)	Effective momentum-transfer cross section
W_es	-sigma_eps*x^2	PDE energy-space flow-velocity coefficient
D_es	(1/3)*(x/sigma_m)*(EoN)^2+ sigma_eps*tau_g*x^2	PDE energy-space diffusion coefficient
tau_e	(2/3)*eps_mean	Electron temperature (eV)
f_M	(2/sqrt(pi))*tau_e^(-3/2)*exp(-x/tau_e)	Maxwellian EEDF
f_init	(2/sqrt(pi))*tau_ei^(-3/2)*exp(-x/tau_ei)	Initial EEDF
C2	-x*sig2(x)*f	Inelastic source/sink
C3	(x+dH3)*sig3(x+dH3)*f3-x*sig3(x)*f	Inelastic source/sink
C4	(x+dH4)*sig4(x+dH4)*f4-x*sig4(x)*f	Inelastic source/sink
C5	(x+dH5)*sig5(x+dH5)*f5-x*sig5(x)*f	Inelastic source/sink
C6	(x+dH6)*sig6(x+dH6)*f6-x*sig6(x)*f	Inelastic source/sink
C7	(x+dH7)*sig7(x+dH7)*f7-x*sig7(x)*f	Inelastic source/sink

NAME	EXPRESSION	DESCRIPTION
C8	(x+dH8)*sig8(x+dH8)*f8-x*sig8(x)*f	Inelastic source/sink
C9	(x+dH9)*sig9(x+dH9)*f9-x*sig9(x)*f	Inelastic source/sink
C10	(x+dH10)*sig10(x+dH10)*f10-x*sig10(x)*f	Inelastic source/sink
C11	(x+dH11)*sig11(x+dH11)*f11-x*sig11(x)*f	Inelastic source/sink
C12	(x+dH12)*sig12(x+dH12)*f12-x*sig12(x)*f	Inelastic source/sink
C13	(x+dH13)*sig13(x+dH13)*f13-x*sig13(x)*f	Inelastic source/sink
C14	(x+dH14)*sig14(x+dH14)*f14-x*sig14(x)*f	Inelastic source/sink
C15	(x+dH15)*sig15(x+dH15)*f15-x*sig15(x)*f	Inelastic source/sink
C16	2*(2*x+dH16)*sig16(2*x+dH16)*f16 -x*sig16(x)*f	Inelastic source/sink
S	C2+C3+C4+C5+C6+C7+C8+C9+C10+ C11+C12+C13+C14+C15+C16	Inelastic collision source
R_norm	-nojac(k16-k2)*sqrt(x)*f	Growth-renormalization term

Here, f3, ..., f16 represent the nonlocal factor $f(\varepsilon + \Delta \varepsilon_k)$ or (in the case of f16) $f(\varepsilon + 2\Delta \varepsilon_k)$ in Equation 5-5; later, you define these as extrusion coupling variables using the auxiliary geometry **Geom2**.

To preserve the normalization of f_0 in the presence of collision processes that do not conserve the electron number, the growth-renormalization term, R_norm, must be added to the right-hand side of the PDE multiplied by the constraint λ defined by Equation 5-6; see Ref. 1 for a discussion. For this model, the electron-number changing processes are the attachment and ionization reactions (no. 2 and 16 in Table 5-1). The nojac operator in the expression for R_norm instructs COMSOL Multiphysics not to include the argument, k16-k2, in the Jacobian computation; see "The Nojac Operator" on page 162 of the COMSOL Multiphysics User's Guide for additional information on this operator.

3 Click **OK** to close the **Global Expressions** dialog box.

Integration Coupling Variables

Compute the integrals appearing in Equations 5-6–5-11 and make them globally available in the model by defining corresponding integration coupling variables.

I With Geom I active, choose

Options>Integration Coupling Variables>Subdomain Variables.

2 Select Subdomain 1, then define integration coupling variables according to the table below. In all cases, use the default settings for **Integration order** (4) and **Global destination** (selected).

NAME	EXPRESSION
f_norm	<pre>sqrt(x)*f</pre>
eps_mean	x^(3/2)*f
mu_N	-(1/3)*gam*fx*x/sigma_m
D_N	(1/3)*gam*f*x/sigma_m
mu_epsN	-(1/(3*eps_mean))*gam*fx*x^2/sigma_m
D_epsN	(1/(3*eps_mean))*gam*f*x^2/sigma_m
k1	gam*sig1(x)*x*f
k2	gam*sig2(x)*x*f
k3	gam*sig3(x)*x*f
k4	gam*sig4(x)*x*f
k5	gam*sig5(x)*x*f
k6	gam*sig6(x)*x*f
k7	gam*sig7(x)*x*f
k8	gam*sig8(x)*x*f
k9	gam*sig9(x)*x*f
k10	gam*sig10(x)*x*f
k11	gam*sig11(x)*x*f
k12	gam*sig12(x)*x*f
k13	gam*sig13(x)*x*f
k14	gam*sig14(x)*x*f
k15	gam*sig15(x)*x*f
k16	gam*sig16(x)*x*f
k1_M	gam*sig1(x)*x*f_M
k2_M	gam*sig2(x)*x*f_M
k3_M	gam*sig3(x)*x*f_M
k4_M	gam*sig4(x)*x*f_M
k5_M	gam*sig5(x)*x*f_M
k6_M	gam*sig6(x)*x*f_M
k7_M	gam*sig7(x)*x*f_M
k8_M	gam*sig8(x)*x*f_M
k9_M	gam*sig9(x)*x*f_M

NAME	EXPRESSION
k10_M	gam*sig10(x)*x*f_M
k11_M	gam*sig11(x)*x*f_M
k12_M	gam*sig12(x)*x*f_M
k13_M	gam*sig13(x)*x*f_M
k14_M	gam*sig14(x)*x*f_M
k15_M	gam*sig15(x)*x*f_M
k16_M	gam*sig16(x)*x*f_M

3 Click OK.

Extrusion Coupling Variables

To implement the nonlocal collision source terms, use extrusion coupling variables.

I With Geom I still selected, choose

Options>Extrusion Coupling Variables>Subdomain Variables.

2 On the Source page, select both subdomains, then enter the following data:

NAME	EXPRESSION	TRANSFORMATION TYPE	SOURCE TRANSFORMATION
f_extr	f	General transformation	х
f_extr_ion	f	General transformation	x

- **3** Click the **Destination** tab.
- 4 From the Geometry list, select Geom2, and from the Level list, select Subdomain.
- 5 From the Variable list, select f_extr. Select Subdomains 1 and 2, then select the Use selected subdomains as destination check box.
- 6 Set the Destination transformation to x+y.
- 7 From the Variable list, select f_extr_ion. Select Subdomain 1 only, then select the Use selected subdomains as destination check box.
- 8 Set the Destination transformation to 2*x+y, then click OK.
- 9 Now with Geom2 selected, choose

Options>Extrusion Coupling Variables>Subdomain Variables.

IO On the **Source** page, select both subdomains, then enter the following data:

NAME	EXPRESSION	TRANSFORMATION TYPE	SOURCE TRANSFORMATION			
f3	f_extr	General transformation	x	х	у	у

II Define the variables f4, ..., f15 using the same data as for f3 in the previous step.

12 Select only Subdomain 1, then enter the following data:

NAME	EXPRESSION	TRANSFORMATION TYPE	SOURCE	TRANSFO	RMATION	
f16	f_extr_ion	General transformation	x	2*x	у	У

I3 Click the **Destination** tab.

- 14 From the Geometry list, select Geom I, and from the Level list, select Subdomain.
- **I5** Select Subdomain 1.
- **I6** From the **Variable** list, select **f3**. Select the **Use selected subdomains as destination** check box, and set the **Destination transformation** for **x** to **x** and that for **y** to dH3.
- 17 Repeat the previous step for each of the variables f4, ..., f16, using as the Destination transformation for y dH4, ..., dH16, respectively.
- **18** Click **OK** to close the **Subdomain Extrusion Variables** dialog box.

PHYSICS MODELING

Space-Independent Equations

To implement the normalization constraint in Equation 5-6, first define a state for λ :

- I From the Physics menu, open the Space-Independent Equations dialog box.
- 2 Enter the following settings; when done, click **OK**.

NAME	EQUATION	INIT	INITT	DESCRIPTION
lam	f_norm-1	1e-5	0	Normalization constraint

3 From the Base unit system list, select None, then click OK.

Subdomain Settings

- I Click the **GeomI** tab.
- 2 From the Physics menu, open the Subdomain Settings dialog box.
- **3** Specify the following PDE coefficients:

SETTINGS	SUBDOMAIN I	SUBDOMAIN 2
Γ	W_es*f-D_es*fx	0
F	S+lam*R_norm	-f
e _a	0	0
d _a	0	0

Note, in particular, the addition to the source term for Subdomain 1 of the growth-renormalization term multiplied by the integral constraint, λ . On

Subdomain 2, which is introduced solely to allow consistent definitions for the extrusion coupling variables, you simply set f_0 to zero.

- 4 Click the lnit tab. Select Subdomain 1, then type f_init in the f(t₀) edit field.
- 5 Click OK.

Boundary Conditions

Implement the boundary conditions in Equation 5-7, substituting x = 500 for infinity.

- I From the Physics menu, open the Boundary Settings dialog box.
- 2 Select the Interior boundaries check box.
- 3 Specify boundary conditions as in the following table; when done, click OK.

SETTINGS	BOUNDARY I	BOUNDARIES 2, 3
Туре	Neumann	Dirichlet
G	0	0
R		-f

MESH GENERATION

- I From the Mesh menu, open the Free Mesh Parameters dialog box.
- 2 On the Global page, type 1.02 in the Element growth rate edit field.
- 3 On the Subdomain page, select Subdomain 1. Set the Maximum element size to 20.
- 4 On the Boundary page, select Boundary 1. Set the Maximum element size to 0.05.
- 5 Click Remesh, then click OK.
- 6 Select Geom2.
- 7 From the Mesh menu, open the Free Mesh Parameters dialog box.
- 8 Click the Custom mesh size option button, then set the Maximum element size to 10.
- 9 Click the Boundary tab. Select Boundaries 1, 2, and 5, then set the Maximum element size to 1.5.
- 10 Click the Point tab. Select Point 1, then set the Maximum element size to 0.05.
- II Click Remesh, then click OK.

COMPUTING THE SOLUTION

- I Click the Solver Parameters button on the Main toolbar.
- 2 From the Solver list, select Parametric.
- **3** In the **Parameter name** edit field, type EoN, and in the **Parameter values** edit field, type [linspace(1.953,49,100) linspace(50,414,25)].

- 4 Click OK.
- 5 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

First, verify the normalization constraint Equation 5-6 for the solution.

- I Click the **GeomI** tab.
- 2 From the Postprocessing menu, choose Subdomain Integration.
- **3** Select Subdomain 1. In the **Expression** edit field, type sqrt(x)*f.
- **4** In the **Solution to use** area, select an arbitrary solution from the **Parameter value** list, then click **Apply**.

The result, which should be 1 (or very close to 1), appears in the message log below the drawing area.

5 Try a few other solutions. When done, click **OK**.

Reproduce the EEDF plots in Figure 5-26 on page 175 with the following steps:

- I From the **Postprocessing** menu, choose **Domain Plot Parameters**.
- 2 On the General page, select 26.664556 from the Solution to use list. Select the Keep current plot check box.
- 3 Click the Line/Extrusion tab.
- 4 In the Expression edit field, type f_M.
- 5 Click the Line Settings button. From the Line color list, select Color, then click the Color button. Select the black swatch, then click OK.
- 6 Click OK to close the Line Settings dialog box.
- 7 In the Domain Plot Parameters dialog box, click Apply.
- 8 In the Expression edit field, type f.
- 9 Click the Line Settings button. From the Line color list, select Color, then click the Color button. Select a red swatch, then click OK.
- **IO** On the **General** page, click the **Title/Axis** button.
- II Click the option button next to the Title edit field; leave the edit field empty.
- 12 Click the option button next to the First axis label edit field, then enter the label Electron energy [eV].
- I3 Click the option button next to the Second axis label edit field, then enter the label EEDF [eV^{-3/2}]. Select the associated Log scale check box.
- 14 Click OK to close the Title/Axis dialog box.

- **I5** In the **Domain Plot Parameters** dialog box, click **OK**.
- **I6** In the figure window, click the **Zoom Window** button, then select the area 0 < x < 20 and $10^{-5} < y < 10^{0}$ in the upper left corner to finish the plot.

Proceed to generate the plots of the ionization-rate coefficients in Figure 5-27:

- I From the Postprocessing menu, choose Global Variables Plot.
- 2 In the Expression edit field, type k16_M, then click the Add Entered Expression button.
- 3 In the x-axis data area, click the lower option button, then click the Expression button. In the Expression edit field, type eps_mean, then click OK to close the X-Axis Data dialog box.
- 4 Click the Line Settings button. From the Line color list, select Color, then click the Color button. Select black, then click OK.
- 5 Select the Legend check box, then click OK to close the Line Settings dialog box.
- 6 Click the Title/Axis button.
- 7 Click the option button next to the Title edit field; leave the edit field empty.
- 8 Click the option button next to the **First axis label** edit field, then enter the label Mean electron energy [eV].
- 9 Click the option button next to the Second axis label edit field, then enter the label Ionization rate coefficient [m³/s]. Select the associated Log scale check box.
- IO Click OK to close the Title/Axis dialog box.
- II Click **Apply** to generate the first graph.
- **12** Select the **Keep current plot** check box.
- I3 In the Quantities to plot list, select k16_M, then click the Remove Selected Quantities to Plot button.
- I4 In the Expression edit field, type k16, then click the Add Entered Expression button.
- 15 Click the Line Settings button. Click the Color button. Select red, then click OK.
- 16 Click Apply to generate the second graph and thus finish the plot.

Next, reproduce Figure 5-28:

- I In the Quantities to plot list, select k16, then click the Remove Selected Quantities to Plot button.
- 2 In the Expression edit field, type k1, then click the Add Entered Expression button.
- 3 Repeat the previous step 15 times for, in turn, k2, ..., k16.

- 4 Click the Line Settings button. Select Cycle from the Line color, Line style, and Line marker lists, then click OK.
- 5 Click the Title/Axis button. Change the Second axis label to Rate coefficients [m³/s], then click OK.
- 6 From the Plot in list, select New figure, then click Apply to generate the plot.

To reproduce the plot of the transport properties in Figure 5-29, do as follows:

- I Click in the Quantities to plot list, then press Ctrl+A to select all entries. Click the Remove Selected Quantities to Plot button.
- 2 In the Expression edit field, type muN, then click the Add Entered Expression button.
- 3 Repeat the previous step for muepsN, DN, and DepsN.
- 4 Click the Title/Axis button. Change the Second axis label to Mobility*N, diffusivity*N. Clear the associated Log scale check box, then click OK.
- **5** From the **Plot in** list, select **New figure**, then click **OK** to generate the plot and close the **Global Variables Plot** dialog box.

Finally, if you have COMSOL Script or MATLAB, you can reproduce the sparsity-pattern plot of the assembled Jacobian matrix in Figure 5-30 in the following way:

- I Press Ctrl+F or choose File>Export>FEM Structure as 'fem'.
- 2 In the script window, enter the following commands:

```
[KC,LC] = femlin(fem);
spy(KC)
```

The sparsity-pattern plot appears in the last opened figure window.

Fluid Dynamics Models

6

This chapter includes examples of fluid dynamics modeling using the incompressible Navier-Stokes equations for stationary and time-dependent fluid flow and the Euler equations for compressible flow in gas dynamics.

Backstep with Argyris Element

This model shows the use of stream functions to model incompressible fluid flow in a divergence-free approach. Instead of conventional p2-p1 Lagrange elements, the model uses C^1 -continuous Argyris elements to discretize the incompressible Navier-Stokes equations. Formulating the flow equations in terms of a stream function eliminates the continuity equation altogether.

Model Definition

The stream function Ψ is introduced through the following definition:

$$(u, v) = \nabla \times \Psi = \left(\frac{\partial \Psi}{\partial y}, -\frac{\partial \Psi}{\partial x}\right).$$

That is, the velocity field is the curl of the stream function. Therefore the divergence of the velocity field is identically zero wherever the stream function is smooth enough.

Introducing the stream function is in itself motivated by the requirement that the divergence of the velocity field should vanish: you can write any divergence-free field as the curl of some vector field, known as a vector potential. The velocity field determines the stream function Ψ only up to the gradient of some function, which means that the stream function corresponding to a given velocity field is not unique.

The Argyris elements belong to the C¹-class of functions, implying that they have continuous first derivatives and function values. Therefore the continuity equation is satisfied weakly everywhere in the domain. Also, pressure has been eliminated from the equations, but you can determine the pressure field afterwards by solving a Poisson-type equation. To get this equation you essentially take the divergence of the momentum equations.

Argyris Element Properties

The advantage of using Argyris elements in the discretization is that the continuity equation is always satisfied. Convergence toward a solution is not inhibited by poor fulfillment of the incompressibility criterion. There are, however, some disadvantages. When using the Argyris elements, 21 degrees of freedom are associated with any triangle compared to 15 for the standard p2-p1 Lagrange elements. Due to different node usage, the total number of degrees of freedom in the model is roughly the same

in the two cases, but the Argyris elements are more tightly connected to one another, which leads to increased memory usage. Further, the Argyris elements are in all respects more complex, thus it takes more time to prepare the problem for the solvers.

Results

Fifth-order Argyris elements provide very high accuracy for smooth solutions. This model has a discontinuity in the gradient of the velocity field at the backstep corner. The Argyris element has degrees of freedoms representing this quantity in its corners. This causes some convergence and accuracy problems, which mean that you must use the parametric solver. The results, however, are of similar accuracy as in the introductory backstep model, "Example: Steady Incompressible Flow" on page 150 in the *COMSOL Multiphysics Modeling Guide*, on a mesh of half the size in number of elements.

Model Library path:

COMSOL_Multiphysics/Fluid_Dynamics/backstep_argyris

Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select 2D from the Space dimension list.
- **2** In the list of application modes, select

COMSOL Multiphysics>PDE Modes>PDE, General Form>Stationary analysis.

- 3 In the Dependent variables edit field, type u v.
- 4 Click OK.

Set the element type to Argyris later on.

OPTIONS AND SETTINGS

I From the **Options** menu, choose **Constants** to set the values needed to parameterize the model.

2 In the **Add/Edit Constants** dialog box enter the following constants, which represent fluid properties and the velocity:

NAME	EXPRESSION
rho0	1.23
Vmean	0.1
eta0	1.79e-5

GEOMETRY MODELING

- I Shift-click the **Rectangle/Square** button on the Draw toolbar to specify a rectangle.
- 2 In the **Rectangle** dialog box, type 0.08 in the **Width** edit field and 0.0101 in the **Height** edit field.
- 3 Click OK.
- 4 Click the **Zoom Extents** button on the Main toolbar.
- **5** Shift-click the **Rectangle/Square** button on the Draw toolbar to specify another rectangle.
- 6 In the **Rectangle** dialog box, type 0.02 in the **Width** edit field and 0.0049 in the **Height** edit field.
- 7 Click OK.
- 8 Press Ctrl+A to select both rectangles.
- 9 Click the Difference button on the Draw toolbar.

PHYSICS SETTINGS

Point Settings

Because the equations contain only derivatives of the real dependent variable—the stream function Ψ —you must set a reference level somewhere to remove the ambiguity in the solution.

- I From the Physics menu, choose Point Settings.
- **2** Select Point 1 and enter the following constraint:

SETTINGS	POINT I
constr	-psi O

Boundary Conditions

- I From the Physics menu, choose Boundary Settings.
- 2 In the Boundary Settings dialog box, click the r tab.

3 Enter the following boundary coefficients:

SETTINGS	BOUNDARY I	BOUNDARIES 2–5	BOUNDARY 6
R(I)	Vmean*6*s*(1-s)-u	- u	0
R(2)	- V	- V	0

4 Click OK.

Subdomain Settings

- I From the Physics menu, choose Subdomain Settings.
- **2** On the Γ and **F** tabs, respectively, enter the PDE coefficients from the following table:

SETTINGS	SUBDOMAIN I
Γ(Ι,Ι)	2*eta0*ux
Γ(1,2)	etaO*(uy+vx)
Γ(2,Ι)	etaO*(uy+vx)
Γ(2,2)	2*eta0*vy
F(1)	rhoO*(u*ux+v*uy)
F(2)	rhoO*(u*vx+v*vy)

- **3** Click the **Element** tab.
- 4 In the Shape edit field, type sharg_2_5('psi').
- 5 In the **gporder** edit field, type 8 8.
- 6 Click OK.

The standard integration order for the Argyris element is 10. In this case only derivatives of psi appear in equations and boundary conditions. Therefore order 8 is sufficient. Reducing the order improves assembly time.

Expression Variables

- I From the **Options** menu, select **Expressions>Subdomain Expressions**.
- 2 In the resulting dialog box, define the following variables.

VARIABLE NAME	LEVEL	DOMAIN	EXPRESSION
u	Subdomain	I	psiy
ux	Subdomain	I	psiyx
uy	Subdomain	I	psiyy
v	Subdomain	I	-psix

VARIABLE NAME	LEVEL	DOMAIN	EXPRESSION
vx	Subdomain	I	-psixx
vy	Subdomain	ļ	-psixy

3 From the **Options** menu select **Expressions>Boundary Expressions**.

4 In the resulting dialog box define the following variables.

VARIABLE NAME	LEVEL	DOMAIN	EXPRESSION
u	Boundary	1:6	psiy
v	Boundary	1:6	-psix

5 Click OK.

MESH GENERATION

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 In the Free Mesh Parameters dialog box, click the Custom mesh size button, then type 2e-3 in the Maximum element size edit field.
- 3 Click the Point tab.
- 4 Select Point 4 and in the Maximum element size edit field type 4e-4.
- 5 Click the **Remesh** button, then click **OK**.

COMPUTING THE SOLUTION

It is possible to use the parametric solver to increase the velocity—and thereby the Reynolds number—step by step. Either give an explicit list of parameter values or let the solver automatically decide an optimum step length. The last option is usually the proper choice when using the parametric solver for difficult problems such as those with medium to high Reynolds number flows.

- I From the Solve menu, choose Solver Parameters.
- 2 In the Solver list, select Parametric.
- **3** Go to the **Parameter** area. In the **Parameter name** edit field, enter Vmean. Then, in the **Parameter values** edit field, type 0.1 0.544.
- 4 Click the Advanced tab.
- 5 In the Null-space function list, select Orthonormal.
- 6 In the Solution form list, select General. This is because the weak form results in expressions that include *u*, which is not a dependent variable when you use Argyris elements.

7 Click OK, then click the Solve button on the Main toolbar to start the simulation.

POSTPROCESSING AND VISUALIZATION

The software displays the solution for the last parameter value. The value of the stream function itself is hardly interesting, but you can take advantage of the fact that contour lines of the stream function are the same as streamlines of the velocity field. The zero-level contour is of particular interest because it immediately shows the extent of the recirculation region and the location of the stagnation point.

- I From the Postprocessing menu, open the Plot Parameters dialog box.
- **2** In the Plot type area, clear the **Surface** check box. Then, select the **Contour** check box.
- **3** Change the value in the **Element refinement** edit field to 10. Before you can enter the number, you must clear the **Auto** check box.
- 4 Click the **Contour** tab, then clear the **Smooth** check box.
- 5 In the **Contour levels** area, click the button under **Vector with isolevels**, and then type 0:0.0005:0.003 in the edit field.





Flow Past a Cylinder

The flow of fluid behind a blunt body such as an automobile is difficult to compute due to the unsteady flows. The wake behind such a body consists of unordered eddies of all sizes that create large drag on the body. In contrast, the turbulence in the thin boundary layers next to the streamlined bodies of aircraft and fish create only weak disturbances of flow.

An exception to this occurs when you place a slender body at right angles to a slow flow because the eddies organize. A von Karman vortex street appears with a predictable frequency and involves the shedding of eddies from alternating sides. Everyday examples of this phenomenon include singing telephone wires and an automobile radio antenna vibrating in an air stream.

From an engineering standpoint, it is important to predict the frequency of vibrations at various fluid speeds and thereby avoid undesirable resonances between the vibrations of the solid structures and the vortex shedding. To help reduce such effects, plant engineers put a spiral on the upper part of high smokestacks; the resulting variation in shape prohibits the constructive interference of the vortex elements that the structure sheds from different positions.

Model Definition

To illustrate how you can study such effects, the following model examines unsteady, incompressible flow past a long cylinder placed in a channel at right angle to the oncoming fluid. The cylinder is offset somewhat from the center of the flow to destabilize what otherwise would be steady-state symmetrical flow. The simulation time necessary for a periodic flow pattern to appear is difficult to predict. A key predictor is the Reynolds number, which is based on cylinder diameter. For low values (below 100) the flow is steady. In this simulation, the Reynolds number equals 100, which gives a developed Karman vortex street, but the flow still is not fully turbulent.

The frequency and amplitude of oscillations are stable features, but flow details are extremely sensitive to perturbations. To gain an appreciation for this sensitivity, you can compare flow images taken at the same time but with such minor differences as are created by different tolerances for the time stepping. It is important to note that this sensitivity is a physical reality and not simply a numerical artifact.

Before calculating the time-varying forces on the cylinder, you can validate the method of computation at a lower Reynolds number using the direct nonlinear solver. This saves time because you can find and correct simple errors and mistakes before the final time-dependent simulation, which requires considerable time.

The viscous forces on the cylinder are proportional to the gradient of the velocity field at the cylinder surface. Evaluating the velocity gradient on the boundary by directly differentiating the FEM solution is possible but not very accurate. The differentiation produces 1st-order polynomials when 2nd-order elements are used for the velocity field. A far better approach is to use a pair of weak-constraint variables to enforce the no-slip condition; at the same time they collect 2nd-order accurate information on the viscous forces.

The drag and lift forces themselves are not as interesting as the dimensionless drag and lift coefficients. These depend only on the Reynolds number and an object's shape, not its size. The coefficients are defined as

$$C_D = \frac{2F_D}{\rho U_{\text{mean}}^2 D}$$
$$C_L = \frac{2F_L}{\rho U_{\text{mean}}^2 D}$$

using the following parameters:

- F_D and F_L are the drag and lift forces
- ρ is the fluid's density
- U_{mean} is the mean velocity
- *D* is the characteristic length, in this case the cylinder's diameter

Results



Figure 6-1 shows the flow pattern resulting from the geometry.

Figure 6-1: A plot of the last time step clearly shows the von Karman path.

The flow around a cylinder is a common benchmark test for CFD algorithms. Various research teams have tried their strengths on this problem using different techniques. Results from some of these experiments have been collected by Schäfer and Turek (Ref. 1), who also used them to compute a probable value for the "real" answer.

Reference

1. M. Schäfer and S. Turek, "Benchmark computations of laminar flow around cylinder", E.H. Hirschel (editor), *Flow Simulation with High-Performance Computers II, Volume 52 of Notes on Numerical Fluid Mechanics*, Vieweg, 1996, pp. 547–566.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow

MODEL NAVIGATOR

- I In the Model Navigator, select 2D from the Space dimension list.
- 2 In the list of application modes, select

COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes, Steady-state analysis. Accept the default entry in the Element list, Lagrange - P₂ P₁ elements.

3 Click OK.

OPTIONS AND SETTINGS

- I From the Options menu, select Axes/Grid Settings.
- 2 In the Axes/Grid Settings dialog box, specify the following settings. To enter the grid spacing, first click the Grid tab and clear the Auto check box.

AXIS		GRID	
x min	-0.3	x spacing	0.2
x max	2.5	Extra x	
y min	-0.3	y spacing	0.05
y max	0.7	Extra y	0.41

3 Click OK.

- 4 From the **Options** menu, select **Constants**.
- **5** Enter the following names, expressions, and descriptions (optional):

NAME	EXPRESSION	DESCRIPTION
rho0	1[kg/m^3]	Density
eta0	1e-3[Pa*s]	Dynamic viscosity
Umax	0.3[m/s]	Maximum inlet velocity

6 Click OK.

These values together with the cylinder having a diameter of 0.1 m lead to a Reynolds number of 20.

GEOMETRY MODELING

From the Draw menu, select the **Rectangle/Square** button. With the mouse, create a rectangle R1 from (0, 0) to (2.2, 0.41).

- **2** From the Draw menu, select the **Ellipse/Circle (Centered)** button. Using the mouse, create a circle C1 with center at (0.2, 0.2) and a radius of 0.05. Use the right mouse button to ensure that you are creating a circle and use the tick mark at y = 0.15 to obtain the correct radius.
- 3 Create the composite object. From the Draw toolbar, click the **Create Composite Object** button. In the **Set formula** edit field enter R1-C1. Click **OK**.



PHYSICS SETTINGS

Application Mode Properties

You must first enable the weak constraints in the **Application Mode Properties** dialog box.

- I From the Physics menu, choose Properties.
- **2** In the **Application Mode Properties** dialog box, go to the **Weak constraints** list and select **On**. From the **Constraint type** list, select **Non-ideal**.
- 3 Click OK.

Boundary Conditions

First set the boundary conditions for the Incompressible Navier-Stokes equations as if the weak constraints did not exist. When you couple a constraint variable to a dependent variable inside the domain, the weak constraint mode analyzes the
constraints specified for the domain variable and replaces them with a weak term involving the constraint variable.

- I From the Physics menu, choose Boundary Settings.
- 2 In the **Boundary Settings** dialog box, enter the following:

SETTINGS	BOUNDARY I	BOUNDARIES 2, 3, 5–8	BOUNDARY 4
Boundary type	Inlet	Wall	Outlet
Boundary condition	Velocity	No slip	Pressure, no viscous stress
u ₀	4*Umax*s*(1-s)		
v ₀	0		
Ро			0

- 3 Click the Weak Constr. tab.
- **4** Select Boundaries 1–4 (the channel boundaries) and clear the **Use weak constraints** check box.
- 5 Click OK.

Subdomain Settings

Define the properties of the fluid and disable the streamline diffusion in the fluid subdomain. Streamline diffusion is not required to obtain good convergence for this model and by disabling it you also increase the accuracy of the solution.

- I From the Physics menu, select Subdomain Settings.
- 2 In the Subdomain Settings dialog box, enter the following material properties:

SETTINGS	SUBDOMAIN I
ρ	rho0
η	eta0

- 3 Click the Artificial Diffusion button.
- 4 In the Artificial Diffusion dialog box, clear the Streamline diffusion check box and click OK.
- 5 Click OK.

MESH GENERATION

I From the Mesh menu, choose Free Mesh Parameters.

- 2 Click the **Custom mesh size** button, enter 0.03 in the **Maximum element size** edit field, enter 1.2 in the **Element growth rate** edit field, and enter 0.1 in the **Mesh curvature factor** edit field.
- 3 Click OK.
- 4 Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar to solve the model.

POSTPROCESSING AND VISUALIZATION

The quantities in the benchmark study are the drag and lift coefficients. These you can calculate easily by integrating the weak constraint variables (which are really Lagrange multipliers corresponding to the viscous forces) and the pressure over the surface of the cylinder. First calculate the drag coefficient:

- I From the Postprocessing menu, choose Boundary Integration.
- 2 Select boundaries 5 to 8, corresponding to the cylinder surface. Enter the drag force expression -1m1*2/(rho0*(2*Umax/3)^2*0.1) in the **Expression** edit field.
- 3 Click Apply.

The result, 5.579263, appears in the message log and agrees well with the interval [5.57, 5.59] given in Ref. 1. Now calculate the lift coefficient.

4 Change the integrand in the **Expression** edit field to $-1m2*2/(rho0*(2*Umax/3)^2*0.1)$.

Also this value, 0.010675, is in perfect agreement with Schäfer and Turek (Ref. 1), who give the interval [0.0104, 0.0110] for acceptable solutions.

5 Click OK.

Time-Dependent Simulation

Now that you know that the machinery works for low Reynolds numbers and steady flow, you can do a time-dependent simulation at Re = 100.

OPTIONS AND SETTINGS

In the **Constants** dialog box change the maximum inflow velocity U_{max} to 1.5[m/s], corresponding to a Reynolds number of 100.

COMPUTING THE SOLUTION

During the first few seconds of the simulation, before the system reaches a state of steady periodic motion, the output is not really interesting. Therefore you can save memory by saving only the value five times per second up to 3.5 seconds, and then fifty times per second for another second and a half. Perform the following steps for the time-dependent simulation of drag and lift:

- I From the Physics select Properties.
- 2 Select Transient from the Analysis type list. Click OK.
- I From the Solve menu, choose Solver Parameters.
- 2 In the Solver list, select Time dependent.
- **3** In the **Times** edit field type [0:0.2:3.5 3.52:0.02:7].
- 4 In the Absolute tolerance edit field type 1e-4.
- 5 Click OK.
- 6 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The difference from the former case can be seen immediately. Downstream of the cylinder, the Karman path is clearly visible (see Figure 6-1 earlier in this discussion).

To see the evolution of the vortex trail from zero velocity until the flow is fully developed, click the **Animation** button on the Plot toolbar on the left-hand side of the user interface. Notice that the time scale of the movie changes after 3.5 seconds.

You can also investigate the forces on the tube as a function of time by assigning the surface integral to a so-called integration-coupling variable:

- I Select the menu item Options>Integration Coupling Variables>Boundary Variables.
- 2 Ctrl-click Boundaries 5 to 8 to select them as a group. As the first Name item type Lift and in the Expression edit field type -lm2*2/(rho0*(2*Umax/3)^2*0.1).
- **3** Clear the **Global destination** check box. This saves memory because you define the integral value to be stored only in one node instead of all the nodes in the model.
- 4 Click the Destination tab and select Point 1. Select the Use selected points as destination check box. Click OK.
- **5** Select the menu item **Solve>Update Model**. This calculates the integral without re-solving the problem.
- 6 From the **Postprocessing** menu, select **Domain Plot Parameters**. Select all the time steps, then click **Plot type: Point plot**. Go to the **Point** tab, select Point 1, and type

Lift in the **Expression** edit field, then click **OK**. The plot clearly shows the oscillations in the lift force.



Figure 6-2: Time evolution of the total lift force on the cylinder.

Finally, you can investigate how suspended particles behave in the flow stream. Consider, for example, small water droplets entering with the air flow.

- I In the **Plot Parameters** dialog box, go to the **Particle Tracing** page.
- 2 Select the Particle tracing plot check box.
- 3 In the Plot type list, select Points.
- 4 In the Predefined forces list, select the Khan and Richardson force (ns).
- 5 On the **Start Points** tab, enter 0 in the **x** edit field and then enter 0.1:0.05:0.3 in the **y** edit field.
- 6 Go to the **Initial Values** page. Clear the **Auto** check box and enter **3.6** in the **Start time** edit field to study the particles only for fully developed flow.
- **7** In the **Initial velocity** edit fields, type u and v to give the particles the same initial velocity as the inflow.
- 8 Go to the Point Settings page, find the When particles leaves the domain list, and select Disappear.
- 9 On the Release page, select Time between releases and specify the time as 0.4.
- **IO** Click the **Advanced** button.
- II Select the Manual tuning of step size check box and enter 0.02 in the Initial time step edit field and 0.01 for Maximum time step.
- 12 Click OK to close the Advanced Parameters window.

13 Go to the **Animate** tab and click **Start Animation**. Remember that the time scale of the movie changes after 3.5 seconds.

Fluid Valve

Introduction

Many different applications such as printers involve the periodic opening and closing of fluid-flow channels. This is generally a difficult problem to model as it implies a moving boundary condition for the part of the geometry that acts as an obstacle for the flow.

An alternative is to use a material property (viscosity), which is easy to vary over time. In this case, specify a very large viscosity (ideally it should be an infinite viscosity) that in effect stops the flow in regions where this large viscosity is present. To simulate the movement of this region of large viscosity, the model includes a logical expression in the subdomain settings.

The model describes a valve where it is possible to direct the flow into one of the two channels. Flow of varying degrees can also occur in both channels during the opening and closing stages.



Model Definition

Figure 6-3: Depiction of the geometry and the operation of the fluid valve. Flow enters from one inlet at the left, but can leave the valve through two outlets at the right. The choice of outlet depends upon the position of the valve pin. In this model, two valve pins oscillate between the following positions: in front of one outlet channel to in front of the other. Sometimes flow is possible through both outlets, depending on the position of the pins.

In this model, the valve pin moves according to a sinusoidal function of time. The inlet velocity is constant.

DOMAIN EQUATIONS

The fluid flow is described by the Navier-Stokes equations:

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0$$

$$\nabla \cdot \mathbf{u} = 0$$
(6-1)

where ρ denotes the density (kg/m³), **u** the velocity vector (m/s), η the viscosity (Ns/m²), and *p* the pressure (Pa). The modeled fluid is air with viscosity 10^{-5} Ns/m² and density 1 kg/m³.

The movement of the valve is described with an analytic expression **pin**, which returns the value of one in the area corresponding to the valve pin and zero elsewhere. The viscosity is then expressed by

$$\eta = \eta_0 + \operatorname{pin} \cdot \eta_{\infty} \tag{6-2}$$

where η_0 is the fluid viscosity, η_∞ is a very large viscosity (ideally infinite), and pin is described by

$$pin = x_{pin} y_{pin} \tag{6-3}$$

where

$$x_{\text{pin}} = (x > x_0)(x < x_1) \tag{6-4}$$

$$y_{\text{pin}} = 1 - (y > y_1) + (y > y_2)$$
 (6-5)

 y_1 and y_2 depend on time, t, according to:

$$y_{1} = -y_{0} + y_{\max} \sin(2\pi t)$$

$$y_{2} = y_{0} + y_{\max} \sin(2\pi t)$$
(6-6)

and where y_0, x_0, x_1 , and y_{max} are fixed in time and describe the size of the valve pin and the amplitude with which the pin moves.

BOUNDARY CONDITIONS

At the inlet, the model uses a fully developed laminar flow. The velocity is set to a parabolic velocity profile with maximum velocity v_{max} equal to 0.25 m/s. At the

outlets, a neutral boundary condition states that the normal component of the stress tensor is zero:

$$\mathbf{n} \cdot \left[-pI + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\right] = 0$$
(6-7)

All other boundaries have a no-slip condition

$$\mathbf{u} = \mathbf{0} \tag{6-8}$$

Results

Figure 6-4 shows the velocity field (modulus of the velocity vector) when the valve is completely open. Firstly, the plot shows that the inlet is smaller than the compartment that it enters, so that some distance is required before the flow reaches another parabolic profile for the main body of the inlet chamber. If you investigate the velocity vector, some recirculation occurring at the corners of the chamber beside the inlet is visible.

Secondly, the structure of the outlet channels leads to a slight thinning toward the middle of the channels. This provides a slight acceleration, and subsequently greater velocity magnitude, in these regions.



Figure 6-4: Velocity field when the valve is completely open.

Figure 6-5 shows the velocity field when the valve is halfway closed from above and fully closed from below. The use of a very large viscosity not only simulates the solidity



of the valve pin but also does a good job of providing a no-slip boundary condition at the edge of the valve pin.

Figure 6-5: Velocity field when the valve is half way closed from above (top) and fully closed from below (bottom).

Figure 6-6 shows a plot of the flow rates (integral of the velocity vector over the outlets) in the upper and lower channels as well as the total flow rate. The figure illustrates the periodic flow due to the periodic motion of the valve pin.



Figure 6-6: The blue line (solid) shows flow rate in the upper branch as a function of time, while the dotted line shows the flow rate in the lower branch. The dashed line shows the sum of the flow in the two branches.

Modeling in COMSOL Multiphysics

The inequalities used to describe the pin motion are, in the COMSOL Multiphysics implementation, replaced by smooth step functions. The smooth step function in this model is flc2hs, which returns a C^2 -continuous step with a given step width.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve

Modeling Using the Graphical User Interface

I Start COMSOL Multiphysics.

- 2 In the Model Navigator on the New page, select 2D from the Space dimension list.
- 3 From the list of application modes, select COMSOL Multiphysics> Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis.
- 4 Click OK.

GEOMETRY MODELING

- I Shift-click the **Rectangle/Square** button in the Draw toolbar.
- 2 Specify the rectangle settings according to the following table; when done, click OK.

PROPERTY	EXPRESSION
Width	0.01
Height	3e-3
Position: Base	Corner
Position: x	0
Position: y	-1.5e-3

- 3 Click the Zoom Extents button on the Main toolbar.
- **4** Draw another rectangle according to:

PROPERTY	EXPRESSION
Width	5e-4
Height	3e-3
Position: Base	Center
Position: x	9e-3
Position: y	0

- 5 Open the Axes/Grid Settings dialog box from the Options menu.
- **6** Enter the following axis settings:

PROPERTY	EXPRESSION
x min	0.009
x max	0.019
y min	0
y max	5e-3

7 Click the Grid page, clear the Auto check box and set y spacing to 5e-4.

8 Click OK.

- 9 Click the 3rd Degree Bézier Curve button in the Draw toolbar.
- **10** Click in the points (0.01, 0.0015), (0.013, 0.0015), (0.013, 0.0045), and (0.016, 0.0045).
- II Switch to the Line tool by clicking the Line button in the Draw toolbar.
- **12** Click the points (0.018, 0.0045), (0.018, 0.0035), and (0.016, 0.0035).
- **13** Switch back to the **3rd Degree Bézier Curve** and click the points (0.013, 0.0035), $(0.013, 5 \cdot 10^{-4})$, and $(0.01, 5 \cdot 10^{-4})$.
- 14 Close the geometry object by right-clicking anywhere in the drawing area.
- **IS** To make the lower branch, use the mirror tool. Make sure that the upper branch geometry object is selected and click the **Mirror** button in the Draw toolbar.
- 16 Let the Point on line settings remain at the default settings. Change the Normal vector from (1, 0) to (0, 1).
- I7 Click OK.
- **I8** Draw a circle with radius $5 \cdot 10^{-4}$ centred in the point (0.01, 0).
- **I9** Click the **Zoom Extents** button.
- **20** Click the **Create Composite Object** button in the Draw toolbar to open the **Create Composite Object** dialog box.
- **21** Clear the **Keep interior boundaries** check box, type R1+C01+C02-C1 in the **Set** formula edit field, and click **Apply**.
- **22** Select the **Keep interior boundaries** check box, type R2+C03 in the **Set formula** edit field, and click **OK**.
- **2** The last step of the geometry modeling is to fillet the left corners. Click the **Fillet**/ **Chamfer** button in the Draw toolbar.
- 24 Select Points 1 and 2, enter 1e-3 in the Radius edit field and click OK.

OPTIONS AND SETTINGS

I Open the **Constants** dialog box from the **Options** menu and enter constants according to the following table. When done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
x0	8.7[mm]	Valve left end point
x1	9.2[mm]	Valve right end point
scale	1e-4	Step function scaling
y_max	1.5[mm]	Valve pin amplitude
eta_0	1e-5[Pa*s]	Fluid viscosity

NAME	EXPRESSION	DESCRIPTION
eta_inf	1[Pa*s]	Large viscosity
rho	1[kg/m^3]	Fluid density
v_max	0.25[m/s]	Maximum inlet velocity

2 From the Options menu, choose Expressions>Scalar Expressions.

3 Enter scalar expressions according to the following table. When done, click OK.

NAME	EXPRESSION
x_range	flc2hs((x-x0)[1/m],scale)*(1-flc2hs((x-x1)[1/m],scale))
y_range	1-flc2hs((y-y1)[1/m],scale)+flc2hs((y-y2)[1/m],scale)
y1	-1.5[mm]+y_max*sin(2*pi*t[1/s])
y2	1.5[mm]+y_max*sin(2*pi*t[1/s])
pin	y_range*x_range
eta	eta_0+eta_inf*pin
pin_vel	y_max*pi*2[1/s]*cos(2*pi*t[1/s])

The reason for the [1/m] entries for the input arguments to flc2hs is to make them dimensionless. Likewise, the input argument to the sin and cos functions must be dimensionless; appending [1/s] to the variable t for time accomplishes this.

- 4 Select Integration Coupling Variables>Boundary Variables from the Options menu.
- 5 Select Boundary 15 from the **Boundary selection** list. Type u_up in the **Name** edit field and u in the **Expression** edit field. Leave the **Integration order** at 4 and **Global destination** active.
- 6 Select Boundary 14 from the **Boundary selection** list. On the second row, type u_down in the **Name** edit field and u in the **Expression** edit field. Leave the **Integration** order at 4 and **Global destination** active.
- 7 Click OK.

PHYSICS SETTINGS

Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu and enter boundary conditions according to the following table. When done, click **OK**.

SETTINGS	BOUNDARY I	BOUNDARIES 5, 6	BOUNDARIES 14, 15	ALL OTHERS
Boundary type	Inlet	Inlet	Open boundary	Wall
Boundary condition	Velocity	Velocity	Normal stress	No slip

SETTINGS	BOUNDARY I	BOUNDARIES 5, 6	BOUNDARIES 14, 15	ALL OTHERS
u ₀	4*v_max*s* (1-s)	0	-	-
v ₀	0	pin_vel	-	-
f ₀	-	-	0	-

Subdomain Settings

Define the properties of the fluid and disable the streamline diffusion in the fluid subdomain. Streamline diffusion is not required to obtain good convergence for this model because the fluid velocity is relatively low.

- I Open the Subdomain Settings dialog box from the Physics menu.
- **2** Select all subdomains.
- 3 Set the Density to rho and the Dynamic viscosity to eta.
- 4 Click the Artificial Diffusion button.
- **5** In the **Artificial Diffusion** dialog box, clear the **Streamline diffusion** check box and click **OK**.
- 6 Click OK.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box from the Mesh menu.
- 2 On the Global page, click the Custom mesh size button, and then set Maximum element size to 0.0004 and Element growth rate to 1.5.
- **3** Click the **Subdomain** tab.
- 4 Select Subdomain 2 from the Subdomain selection list. Set Maximum element size to 0.0001.
- 5 Click **Remesh** and then **OK**.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box from the Solve menu.
- 2 In the Time stepping area, set Times to 0:0.02:1.5, Relative tolerance to 0.001, and Absolute tolerance to 0.0001.
- 3 Click OK.
- 4 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To create the image in Figure 6-4, do the following steps.

- I Open the Plot Parameters dialog box from the Postprocessing menu.
- 2 On the **General** page, clear the **Geometry edges** check box and make sure that the **Surface** check box is selected.
- **3** On the Surface page, select Velocity field from the Predefined quantities list on the Surface Data tab and click Apply.

To create the images in Figure 6-5 do the following steps:

- 4 On the General page, select the time step at 0.22 from the Solution at time list.
- **5** Select the **Contour** plot type.
- 6 On the **Contour** page, type eta in the **Expression** edit field.
- 7 In the Contour levels frame, select the Vector with isolevels and type linspace(1e-3,1.1e-3,10).
- 8 In the Contour color area, select Uniform color, click the Color button, and select white color. Click OK to close the Contour Color dialog box. Clear the Color scale check box.
- 9 Click Apply in the Plot Parameters dialog box.

To create the top image, repeat the above steps but for time step 0.92.

Use the following steps to create Figure 6-6:

- I Open the Domain Plot Parameters dialog box from the Postprocessing menu.
- 2 On the General page, select the Keep current plot check box.
- 3 Click the **Point** tab and select Point 1 from the **Point selection** list.
- **4** Type u_up in the **Expression** edit field.
- 5 Click the Line Settings button and set Line color to a blue color by first selecting Color from the Line color list and then clicking the Color button. Select Solid line from the Line style list. Click OK.
- 6 Click Apply.
- 7 Type u_down in the Expression edit field.
- 8 Click the Line Settings button and set Line color to a red color and Line style to Dotted line. Click OK.
- 9 Click Apply.
- **IO** Repeat for the expression u_down+u_up with a dashed, black line.

Micromixer

Introduction

This example studies a laminar static mixer with two parallel sets of split-reshaperecombine mixing elements. Figure 6-7 shows the geometry of a single mixing element.



Figure 6-7: The micromixer splits the incoming fluid in the direction perpendicular to the interface separating the two fluid layers. After recombining them, the mixer stacks the two flows on top of each other, resulting in four fluid layers.

Each mixing element doubles the number of fluid layers, resulting in a fast mixing process. This technique is suited for laminar flow mixing and has small pressure losses. In this model, the mixing structure consists of two parallel sets of mixing elements, where each set is two elements long. You measure the mixing quality with the relative variance of the concentration profile, S, calculated as

$$S = \frac{s_x}{s_{\text{inlet}}}$$
$$s_x = \int_{K_x} (c - \bar{c})^2 dA$$

where K_x is the *yz*-plane intersecting the mixing structure at coordinate *x*, and *c* is the mean concentration.

DOMAIN EQUATIONS

The fluid flow is described by the Navier-Stokes equations

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0$$
$$\nabla \cdot \mathbf{u} = 0$$

where ρ denotes the density (kg/m³), **u** the velocity (m/s), η denotes the viscosity (N·s/m²), and *p* represents the pressure (Pa). The modeled fluid is water with a viscosity of 1·10⁻³ N·s/m² and a density of 1000 kg/m³.

The mass flux is given by diffusion and convection, and the resulting mass balance is

$$\nabla \cdot (-D\nabla c + c\mathbf{u}) = 0$$

where *D* denotes the diffusion coefficient (m^2/s) and *c* gives the concentration (mol/m^3) . The modeled species is bicarbonate ions with a diffusion coefficient of $1.49 \cdot 10^{-9} \text{ m}^2/\text{s}$.

BOUNDARY CONDITIONS

At the inlet the model assumes fully developed laminar flow. It sets the velocity to a parabolic profile with a mean velocity of 0.01 m/s. At the outlet, the model sets the pressure to zero. All other boundaries have a no-slip condition

 $\mathbf{u} = 0.$

The inlet concentration has a discontinuous profile where the upper half has a concentration of 27 mol/m³ and the lower half is pure water. The boundary condition is defined such that

$$c|_{\text{inlet}} = \begin{cases} c_0 & z > 0\\ 0 & z \le 0. \end{cases}$$

Results

Figure 6-8 shows the final concentration profile.



Figure 6-8: Following three split-reshape-recombine cycles, the outflow has eight fluid layers.

Using boundary integration to calculate the relative variance of the concentration at the inlet and outlet, this setup reaches a mixing quality of 0.31 (for perfect mixing this value would be 0, and a value of 1 means no mixing at all). To improve this quality, add mixing elements.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/micromixer

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

I Open the Model Navigator, go to the Space dimension list and select 3D. Click the Multiphysics button.

2 Select the application mode

COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Steady-state analysis, then click the Add button.

- 3 Select COMSOL Multiphysics>Convection and Diffusion>Convection and Diffusion. Click the Add button once again.
- 4 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu select **Constants**.
- **2** Define the following names, expressions, and (optionally) descriptions:

NAME	EXPRESSION	DESCRIPTION
rho	1e3[kg/m^3]	Density
eta	1e-3[Pa*s]	Dynamic viscosity
U_mean	0.01[m/s]	Mean inlet velocity
c0	27[mol/m^3]	Inlet concentration
D	1.49e-9[m^2/s]	Diffusion coefficient

3 Click OK.

GEOMETRY MODELING

The geometry consists of a rectangular channel that is split and recombines several times. Start by defining a work plane.

- I From the **Draw** menu select **Work-Plane Settings**. Click **OK** to select the default **x**-**y** work plane at z = 0.
- 2 Select the menu item Options>Axes/Grid Settings.
- **3** Verify that you are on the **Axis** page. Enter the following settings:

PROPERTY	EXPRESSION
x min	-1e-3
x max	3e-3
y min	-1e-3
y max	1e-3

- 4 Go to the Grid page. Clear the Auto check box, then in both the x spacing and y spacing edit fields enter 1e-4.
- 5 Click OK.

Constructing a Single Mixing Element

- I Go to the Draw toolbar on the left side of the user interface and click the Line button. Draw a seven-sided polygon in the drawing area by clicking, in order, the coordinates (0, 0), (0, 0.5 ⋅ 10⁻³), (0.7 ⋅ 10⁻³, 0.7 ⋅ 10⁻³), (1.3 ⋅ 10⁻³, 0.7 ⋅ 10⁻³), (2 ⋅ 10⁻³, 0.5 ⋅ 10⁻³), (2 ⋅ 10⁻³, -0.5 ⋅ 10⁻³), (1 ⋅ 10⁻³, 0.1 ⋅ 10⁻³), and (0, 0). Finally, click the right mouse button to create the solid object. (Soon you will learn about an alternative method for creating polygons.)
- 2 From the **Draw** menu choose **Extrude**. In the **Distance** edit field enter 2e-3, then click **OK**.
- 3 Click the Move button on the Draw toolbar. Set the Displacement: z to -1e-3, then click OK.
- **4** Press Ctrl+C and then Ctrl+V to make a copy of the extruded object and paste it. Let all **Displacements** equal 0, then click **OK**.
- 5 On the Draw toolbar click the Rotate button. Set the Rotation angle to 90, then set the Rotation axis direction vector to x, y, and z values of 1, 0, and 0, respectively. Click OK.
- 6 Return to the Draw toolbar and again click the Rotate button. Set the Rotation angle to 180. This time go to the Point on rotation axis area, and in the x, y, and z edit fields enter 1e-3, 0, and 0, respectively. Click OK.
- **7** Press Ctrl+A to select both objects, then click the **Intersection** button on the Draw toolbar.
- 8 Make a copy of the object by pressing Ctrl+C, then Ctrl+V, and choosing OK.
- **9** Go to the Draw toolbar and click the **Rotate** button. Set the **Rotation angle** to 180, then set the **Rotation axis direction vector** to **x**, **y**, and **z** values of 1, 0, and 0, respectively. Click **OK**.

Connecting a Second Mixing Element in the Series

- I Press Ctrl+A to select both objects.
- **2** Go to the Draw toolbar and click the **Mirror** button and in the **Normal vector** row enter values for **x**, **y**, and **z** of 0, 1, and 0, respectively. Click **OK**.
- 3 Click the Move button on the Draw toolbar. Set the Displacement: x to 2.5e-3. Click OK.

Making Square Channel Connections Between the Mixing Elements

- I From the **Draw** menu select the **Geom2** work plane.
- 2 From the Draw menu select the **Rectangle/Square** button and note that the cursor changes to a crosshair (alternatively, select **Draw>Draw Objects>Rectangle/Square**).

Using the mouse, drag the cursor to create a rectangle with its base corner at $(-0.5 \cdot 10^{-3}, -0.5 \cdot 10^{-3})$, a width of $0.5 \cdot 10^{-3}$, and a height of $1 \cdot 10^{-3}$.

- 3 Select the menu item **Draw>Extrude**. In the **Distance** edit field enter 1e-3, then click **OK**.
- 4 On the Draw toolbar click the Move button. Set the Displacement: z to -0.5e-3.
- 5 On the Draw toolbar click the **Array** button. Change the **Displacement: x** to 2.5e-3 and the **Array size: x** to 3. Click **OK** to create the array.

Getting Two Mixing Channels in Parallel

- I Press Ctrl+A to select all the objects.
- 2 From the Draw toolbar select Move. Set the Displacement: y to 1e-3. Click OK.
- **3** Press Ctrl+C and Ctrl+V to copy and paste the objects. Set the **Displacement: y** to -2e-3, then click **OK**.

Connecting the Two Parallel Channels to a Single Channel at Both Ends

- I From the Draw menu select the Geom2 work plane.
- 2 On the Draw toolbar, Shift-click the **Rectangle/Square** button. In the **Rectangle** dialog box create a rectangle with its base corner at $(-3.5 \cdot 10^{-3}, -0.7 \cdot 10^{-3})$, a width of 10^{-3} , and a height of $1.4 \cdot 10^{-3}$.
- 3 From the Draw menu choose Extrude. Set the Distance to 1.4e-3, then click OK.
- 4 Go to the Draw toolbar and click the Move button. Set the Displacement: z to -0.7e-3, then click OK.
- **5** Press Ctrl+C and Ctrl+V to copy and paste the objects; set the **Displacement: x** to 10.5e-3, then click **OK**.
- 6 Click on the tab at the top of the drawing area to return to the Geom2 work plane. On the Visualization/Selection toolbar on the left side of the user interface choose Projection of All 3D Geometries.
- 7 Draw a four-sided polygon with corners at the points (-2.5·10⁻³, 0), (-0.5·10⁻³, 0.5·10⁻³), (-0.5·10⁻³, 1.5·10⁻³), and (-2.5·10⁻³, 0.7·10⁻³). To do so, select the menu item Draw>Specify Objects>Line (or Shift-click the Line button in the Draw toolbar). In the resulting dialog box find the Style list and select Closed polyline (solid). In the x edit field enter the sequence of x-coordinates from the coordinate pairs just given, then do the same for the y-coordinates in the y edit field. Click OK.
- 8 Next, extrude the polygon: select the menu item **Draw>Extrude**, then in the **Distance** edit field type 2e-3. Click **OK**.

9 Move the extruded polygon. From the Draw toolbar select the Move tool. You default to the Displacement area, and in the z edit field enter -1e-3.

Setting Up a Second Work Plane

- From the Draw menu select Work-Plane Settings, then click Add. A new work plane, Geom3, appears. Select the z-x plane and click OK.
- 2 On the Visualization/Selection toolbar choose Projection of All 3D Geometries.
- 3 Click the Zoom Extents button on the Main toolbar.
- 4 From the **Options** menu open the **Axes/Grid Settings** dialog box. On the **Grid** page clear the **Auto** check box. In both the **x spacing** and **y spacing** edit fields enter 1e-4.
- **5** Using the **Closed polyline (solid)** technique already described, draw a four-sided polygon with corners at the points $(-0.7 \cdot 10^{-3}, -2.5 \cdot 10^{-3}), (0.7 \cdot 10^{-3}, -2.5 \cdot 10^{-3}), (0.5 \cdot 10^{-3}, -0.5 \cdot 10^{-3}), and <math>(-0.5 \cdot 10^{-3}, -0.5 \cdot 10^{-3}).$
- 6 Select Draw>Extrude, then in the Distance edit field type 2e-3. Click OK.
- **7** Select the two extruded geometries (EXT9 and EXT10). Go to the Draw toolbar and click the **Intersection** button.
- 8 Go back to the Draw toolbar and click the Mirror button. Set the x, y, and z values of the Normal vector to 0, 1, and 0, respectively. Click OK.

Doubling the Number of Fluid Layers

When building the outflow connections, notice that you can double the fluid layers one more time by guiding the channels properly.

- I Click the appropriate tab at the top of the drawing area to return to the Geom2 work plane.
- **2** Using the **Closed polyline (solid)** technique already described, draw a five-sided polygon with corners at the points $(5 \cdot 10^{-3}, 0.5 \cdot 10^{-3}), (6 \cdot 10^{-3}, 0.1 \cdot 10^{-3}), (7 \cdot 10^{-3}, -0.7 \cdot 10^{-3}), (7 \cdot 10^{-3}, 0.7 \cdot 10^{-3}), and <math>(5 \cdot 10^{-3}, 1.5 \cdot 10^{-3})$.
- 3 Select Draw>Extrude, then in the Distance edit field type 2e-3. Click OK.
- 4 Move the extruded polygon. From the Draw toolbar select the Move tool. You default to the Displacement area, and in the z edit field enter -1e-3. Click OK.
- 5 Go to the Geom3 work plane.
- **6** Using the **Closed polyline (solid)** technique already described, draw a six-sided polygon with corners at the points $(0.5 \cdot 10^{-3}, 5 \cdot 10^{-3}), (0.8 \cdot 10^{-3}, 6 \cdot 10^{-3}), (0.7 \cdot 10^{-3}, 7 \cdot 10^{-3}), (0, 7 \cdot 10^{-3}), (0.1 \cdot 10^{-3}, 6 \cdot 10^{-3}), and (-0.5 \cdot 10^{-3}, 5 \cdot 10^{-3}).$
- 7 Select Draw>Extrude, then in the Distance edit field type 3e-3. Click OK.

- 8 From the Draw toolbar select the Move tool. You default to the Displacement area, and in the y edit field enter -1e-3. Click OK.
- **9** Select the EXT9 and EXT10 geometries; on the Draw toolbar click the **Intersection** button.
- **10** Press Ctrl+C and Ctrl+V to make a copy of the new object. Let all **Displacements** equal **0**. Click **OK**.
- II Go to the Draw toolbar and click the Rotate button. Set the Rotation angle to 180, then set the Rotation axis direction vector to x, y, and z values of 1, 0, and 0, respectively. Click OK.
- 12 Press Ctrl+A to select all the geometry objects.
- **I3** From the Draw toolbar choose **Create Composite Object**. Clear the **Keep interior boundaries** check box, then click **OK**.

This completes the geometry-modeling stage. The resulting geometry should look like that in the figure below.



Model geometry.

PHYSICS SETTINGS-INCOMPRESSIBLE NAVIER-STOKES

Boundary Conditions

- I From the Multiphysics menu select I Incompressible Navier-Stokes (ns).
- 2 From the **Physics** menu open the **Boundary Settings** dialog box. Enter settings from the following table; when done, click **OK**.

SETTINGS	BOUNDARY I	BOUNDARY 136	ALL OTHERS
Boundary type	Inlet	Outlet	Wall
Boundary condition	Velocity	Pressure, no viscous stress	No slip

SETTINGS	BOUNDARY I	BOUNDARY 136	ALL OTHERS
u ₀	(9/4)*U_mean*16*s1* (1-s1)*s2*(1-s2)	-	-
v ₀	0	-	-
w ₀	0	-	-
Po	-	0	-

Subdomain Settings

- I Select the menu item Physics>Subdomain Settings.
- 2 Select Subdomain 1. Set the Density to rho and the Dynamic viscosity to eta.
- **3** Click the **Init** tab and then select Subdomain 1. Set the following initial conditions; when finished, click **OK**.

VARIABLE	VALUE
u(t ₀)	U_mean
v(t ₀)	0
w(t ₀)	0
p(t ₀)	0

PHYSICS SETTINGS—CONVECTION AND DIFFUSION

Boundary Conditions

- I From the Multiphysics menu select 2 Convection and Diffusion (cd).
- 2 Select the menu item Physics>Boundary Settings. Enter the following settings, then click OK:

SETTINGS	BOUNDARY I	BOUNDARY 136	ALL OTHERS
Туре	Concentration	Convective flux	Insulation/Symmetry
c ₀	c0*(z<0)		

Subdomain Settings

- I From the Physics menu open the Subdomain Settings dialog box.
- 2 Select all the subdomains, then enter coefficients from the following table:

NAME	EXPRESSION
D (isotropic)	D
R	0
u	u

NAME	EXPRESSION
v	v
w	w

- 3 Click the Artificial Diffusion button. In the resulting dialog box, select the Streamline diffusion check box. Click OK.
- 4 Click the **Init** tab. In the **c(t**₀) edit field enter c0.
- 5 Click OK.

MESH GENERATION

- I From the Mesh menu select Free Mesh Parameters.
- 2 Click the **Custom mesh size** button and enter 3e-4 in the **Maximum element size** edit field.
- 3 Click Remesh, then click OK.

COMPUTING THE SOLUTION

Because the fluid's properties do not change with concentration, this example solves the Navier-Stokes equations independently.

- I Click the Solver Manager button on the Main toolbar.
- 2 Go to the Solve For page. Select Incompressible Navier-Stokes (ns), then click OK.
- 3 Click the Solver Parameters button on the Main toolbar.
- **4** On the **General** page, select **Geometric multigrid** from the **Linear system solver** list; then click **OK**. This also sets up the Vanka smoother with pressure as the so-called Vanka variable.
- 5 Click the Solve button on the Main toolbar to calculate the flow field.

Solving the Mass Balance

With the flow field computed, it is time to turn to the mass balance equation. The example solves the mass balance on a finer mesh so it can resolve the high gradients in the interface between the fluid layers.

- I From the Mesh menu open the Free Mesh Parameters dialog box.
- 2 Click the **Custom mesh size** button and in the **Maximum element size** edit field enter 1.25e-4.
- 3 Click Remesh, then click OK.

Adjust the solver settings to solve for the mass balance. Because the convection-diffusion equation is linear, use the linear solver.

- I From the Solve menu open the Solver Manager dialog box.
- 2 Click the Solve For tab. Select Convection and Diffusion (cd).
- 3 Click the Initial Value tab, then click the Store Solution button.
- 4 In the Initial value area, click the Stored solution option button. Click OK.
- 5 From the Solve menu open the Solver Parameters dialog box.
- 6 From the Linear system solver list choose GMRES. Click OK.
- 7 Compute the final solution by clicking the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

Follow these instructions to reproduce the plot in Figure 6-8.

- I From the **Postprocessing** menu open the **Plot Parameters** dialog box.
- **2** On the **General** page, locate the **Element refinement** label, clear the **Auto** check box, then in the associated edit field enter **2**.
- 3 In the Plot type area clear the Slice check box and select the Boundary check box.
- 4 Go to the Boundary page. In the Predefined quantities list select Convection and Diffusion (cd)>Concentration, c.
- 5 Click OK.

Finally calculate the mixing quality.

- I From the **Postprocessing** menu open the **Boundary Integration** dialog box.
- 2 Select Boundary 1 (the inlet), then in the Expression edit field enter (c-c0/2)^2.
- **3** Click **Apply** to perform the integration. The result appears in the report log at the bottom of the user interface.
- **4** Select Boundary 136 (the outlet). Click **OK** to perform the integration and close the dialog box.
- **5** Dividing the outlet value by the inlet value gives a measure of the mixing quality.

Shock Tube

Shock waves arise from sudden jumps in gas properties such as temperature or pressure. We generally associate shock waves with violent processes such as supersonic flight or explosion blasts. However, even the slow movement of a piston in a tube can create flows that eventually turn into shock waves due to the inherent dynamics of the conservation of mass, momentum, and energy.

Discontinuities in flows were the topic of debate in the late 19th century, and even Lord Rayleigh at some point concluded they could not exist. What finally settled the issue was photographic evidence from a shock-tube experiment such as that described here.

Introduction

A shock tube is a device for studying shock waves. Prior to starting an experiment, a diaphragm inside the tube blocks any flow. You then increase the pressure on one side, for example, by using a compressor, and start the flow by rupturing the diaphragm. The gas then expands down the other half of the tube. Through optical means you can observe the flow and model the action of shocks. A pressurized tube of this type can store substantial amounts of energy, allowing for the study of quite violent flows—albeit for only very short periods.

Model Definition

The effects of viscosity and heat conduction are small for the time scales of interest here. Therefore the Euler equations of gas dynamics define the flow as movement of a compressible inviscid gas in the tube according to

$$\rho_t + u\rho_x + u_x\rho = 0$$
$$u_t + uu_x + \frac{p_x}{\rho} = 0$$
$$p_t + up_x + a^2\rho u_x = 0$$

where u is the velocity, ρ is the density, and p is the pressure.

The speed of sound *a* in a polytropic gas is given by:

$$a^2 = \gamma \frac{p}{\rho}$$

Here γ gives the ratio of specific heats at constant pressure to constant volume; for a diatomic gas such as air, γ is 7/5. Thus you can write the equations as

$$0 = -(\rho_t + u\rho_x + u_x\rho)$$
$$0 = -(u_t + uu_x + \frac{p_x}{\rho})$$
$$0 = -(p_t + up_x + \gamma pu_x)$$

Modeling in COMSOL Multiphysics

A clever way to conserve on model development time and computational cost is to solve this problem in a 2D geometry, using the *y*-coordinate as time. To do so, set up a rectangular domain in the ranges -1.5 < x < 1.5 and 0 < y < 1. As initial conditions, use the following values:

$$\begin{cases} u = 0 \\ \rho = \begin{cases} 2 & x < 0 \\ 1 & x > 0 \end{cases} \\ p = \begin{cases} 2 & x < 0 \\ 1 & x > 0 \end{cases}$$

As these equations are numerically difficult to solve, stabilization with streamline diffusion facilitates convergence of the numerical scheme.

To derive the streamline-diffusion contributions, create a test function by taking each of the original equations and replacing the first-order derivatives with the corresponding test functions. Multiply the results by δh and the original equations to obtain equations for the streamline-diffusion contributions

$$-\delta h(\hat{\rho}_t + u\hat{\rho}_x + u_x\rho)(\rho_t + u\rho_x + u_x\rho)$$
$$-\delta h(\hat{u}_t + u\hat{u}_x + \frac{\hat{p}_x}{\rho})(u_t + uu_x + \frac{p_x}{\rho})$$
$$-\delta h(\hat{p}_t + u\hat{p}_x + \gamma \hat{p}\hat{u}_x)(p_t + up_x + \gamma pu_x)$$

The above equations denote test functions using the "hat" symbol. This streamline diffusion is of the Petrov-Galerkin type, which is one of the predefined types of artificial diffusion that you find, for example, in the Incompressible Navier-Stokes application mode.

In the model, you enter the Euler equations using a general form PDE with three dependent variables for pressure, density, and velocity. Use a weak term to add the streamline diffusion.

To resolve the shock wave, use the adaptive mesh generation in COMSOL Multiphysics.

Results

The plots below show distance across the tube's diaphragm on the horizontal axis and time along the vertical axis, where temperature and pressure are color coded. At t = 0 you can see constant states on both sides of the membrane, which is centered at x = 0. Specifically, in Figure 6-9, pressure is high to the left and low to the right; in Figure 6-10, temperature is uniform throughout the tube.



Figure 6-9: The pressure distribution in the shock tube. The y-axis represents time.



Figure 6-10: The nondimensional temperature distribution. The y-axis represents time.

The plots show the typical flow features of shock-tube problems. There are two possible types of jump solutions. At the time of a shock, all variables have jump discontinuities, and the second type of jump solution is a *contact surface* where pressure is continuous.

This case shows a shock traveling right, a contact surface moving more slowly to the right, and finally an expansion fan receding to the left into the denser, undisturbed gas.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/shock_tube

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select 2D from the Space dimension list.
- 2 Click the Multiphysics button.

- 3 Click the Add Geometry button.
- **4** Keep **2D** in the **Space dimension** list and type x time z in the **Independent variables** edit field.
- 5 Click OK.
- 6 In the COMSOL Multiphysics>PDE Modes folder, select PDE, General Form.
- 7 Type rho u p in the Dependent variables edit field.
- 8 Click Add and then click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- **2** In the **Constants** dialog box, define the following constants with names and expressions:

NAME	EXPRESSION
gamma	1.4
delta	1

- 3 Click OK.
- 4 From the Options menu, choose Axes/Grid Settings.
- **5** Enter axis settings:

AXIS	
x min	-1.5
x max	1.5
time min	-0.5
time max	1.5

6 Click OK.

GEOMETRY MODELING

- I Click the Rectangle/Square button in the Draw toolbar.
- **2** Draw a rectangle with opposite corners in the points(-1.5, 0) and (1.5, 1).

PHYSICS SETTINGS

Expression Variables

I On the **Options** menu, point to **Expressions** and then click **Subdomain Expressions**.

2 Select Subdomain 1 and enter the following expression variables:

PROPERTY	VALUE
FI	-(rhotime+u*rhox+rho*ux)
F2	-(utime+u*ux+px/rho)
F3	-(ptime+u*px+gamma*p*ux)

3 Click OK.

Boundary Conditions

I From the Physics menu, choose Boundary Settings.

2 In the **Boundary Settings** dialog box, enter the following settings:

SETTINGS	BOUNDARIES I, 4	BOUNDARY 2	BOUNDARY 3
G(I)	0	0	0
G(2)	0	0	0
G(3)	0	0	0
R(I)	0	(x<0)+1-rho	0
R(2)	- u	- u	0
R(3)	0	(x<0)+1-p	0

3 Click OK.

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

2 In the Subdomain Settings dialog box, enter the following settings:

PROPERTY	VALUE
Γ(Ι)	0 0
Γ(2)	0 0
Г(3)	0 0
F(1)	F1
F(2)	F2
F(3)	F3

3 Click the **Init** tab and set the following initial conditions:

PROPERTY	VALUE
rho(t ₀)	1
u(t ₀)	0
P(t ₀)	1

- 4 Click the Weak tab.
- **5** In the **weak** edit field, type the following streamline diffusion contributions as one single line, separating each contribution with a space:

```
delta*h*(rhotime_test+u*rhox_test+rho*ux_test)*F1
delta*h*(utime_test+u*ux_test+px_test/rho)*F2
delta*h*(ptime_test+u*px_test+gamma*p*ux_test)*F3
```

6 Click OK.

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

- I From the Solve menu, choose Solver Parameters.
- 2 Select the Adaptive mesh refinement check box.
- **3** Click the **Advanced** tab.
- 4 In the Scaling of variables area, select None in the Type of scaling list.
- 5 Click OK.
- 6 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

- I Open the Plot Parameters dialog box.
- 2 On the Surface tab, visualize the pressure by selecting PDE, General Form (g)>p from the Predefined quantities list on the Surface Data tab.
- 3 Click Apply.

This generates the plot in Figure 6-9.

- 4 Similarly, to visualize the nondimensional temperature, type p/rho in the **Expression** edit field on the **Surface Data** tab.
- 5 Click OK.

This closes the Plot Parameters dialog box and generates the plot in Figure 6-10.

Terminal Falling Velocity of a Sand Grain

Introduction

The first stop for polluted water entering a water work is normally a large tank, where large particles are left to settle. More generally, gravity settling is an economical method of separating particles. If the fluid in the tank is moving at a controlled low velocity, the particles can be sorted in separate containers according to the time it takes for them to reach the bottom.

This model simulates a spherical sand grain falling in water. The grain accelerates from standstill and rapidly reaches its terminal velocity. The results agree very well with experimental studies. The model is an axially symmetric fluid-flow simulation in a moving coordinate system, coupled to an ordinary differential equation (ODE) describing the grain's motion.

Model Definition

The model couples the flow simulation in cylindrical coordinates with an ODE for the force balance of the particle. Due to axial symmetry, you can model the flow in 2D instead of 3D. The figure below shows the modeling domain.



DOMAIN EQUATIONS

The fluid flow is described by the Navier-Stokes equations

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{F}$$
$$\nabla \cdot \mathbf{u} = 0$$

where ρ denotes the density (kg/m³), **u** the velocity vector (m/s), η the viscosity (Ns/m²), and *p* the pressure (Pa). The fluid is water with a viscosity of $1.51 \cdot 10^{-3}$ Ns/m² and a density of 1000 kg/m³. The problem is solved in the accelerating reference system of the sand grain. This means that the volume force density **F** is given by:

$$F_r = 0, \qquad F_z = -\rho(a+g)$$

where $a \text{ (m/s}^2)$ is the acceleration of the grain and $g = 9.81 \text{ m/s}^2$ is the acceleration due to gravity. The ODE that describes the force balance is:

$$m\ddot{x} = F_g + F_z$$
where m (kg) denotes the mass of the particle, x (m) the position of the particle, F_g (N) the gravitational force, and F_z the z-component of the force that the water exerts on the sand grain. The gravitational force is given by:

$$F_g = -\rho_{\text{grain}} V_{\text{grain}} g$$

where V_{grain} (m³) is the volume of the sand grain and ρ_{grain} (kg/m³) its density. The force that the water exerts on the grain is calculated by integrating the normal component of the stress tensor over the surface of the particle. Because the model is axially symmetric, the force only has a nonzero *z*-component:

$$F_{z} = 2\pi \int_{S} r\mathbf{n} \cdot [-pI + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^{T})] dS$$

where r (m) is the radial coordinate and **n** is the normal vector on the surface of the grain.

The problem is solved using the initial values $u_0 = v_0 = x_0 = \dot{x_0} = 0$.

BOUNDARY CONDITIONS

At the surface of the sphere, the velocity relative the sphere is zero. Therefore, a *no-slip* condition, $\mathbf{u} = \mathbf{0}$, is used. The velocity at the inlet of the fluid domain is set to the falling velocity: $\mathbf{u} = (\mathbf{0}, \dot{x})$. Symmetry (or the *slip* condition), $\mathbf{n} \cdot \mathbf{u} = \mathbf{0}$, is set at the *z*-axis, and a *neutral* condition, $\mathbf{n} \cdot [-pI + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] = \mathbf{0}$, describes the outlet.

Results

The following series of snapshots are of the velocity field, from a moment just after the sand grain is released until it is approaching steady state. Notice the recirculation





Figure 6-11: The velocity field around the sand grain at a series of times.

Figure 6-12 below shows the falling velocity of the grain as a function of time. The



Figure 6-12: Falling velocity (m/s) of the grain versus time. After the solution time of 1 s, the velocity approaches the terminal velocity.

terminal velocity equals 0.291 m/s. When this state is reached, the gravity and the forces from the water cancel out. Figure 6-13 below shows the forces on the sand grain.



Figure 6-13: The forces on the sand grain. The force that the water exerts on the sphere (upper line) increases as the grain gains speed. The gravity force (lower line) remains the

same, and the total force (middle line) tends toward zero as the solution approaches steady state.



Figure 6-14 below shows the velocity field at steady state.

Figure 6-14: The velocity field at steady state. Note that the velocities are plotted in the reference system of the sand grain.

Several approximate equations have been proposed for the terminal velocity of a sphere falling in a fluid. Ref. 1 cites the following expression for the total force that the fluid exerts on the sphere, as a function of the velocity:

$$F = \frac{\pi}{4} d^2 \rho v^2 (1.84 \text{Re}^{-0.31} + 0.293 \text{Re}^{0.06})^{3.45}$$

where *d* (m) is the diameter of the sphere, ρ (kg/m³) the fluid density, *v* (m/s) the velocity, and Re = $(\rho v d)/\mu$ is the Reynolds number, with μ (Ns/m²) being the viscosity of the fluid. The gravity force is given analytically as $F_g = \pi d^3(\rho - \rho_s)g/6$, where ρ_s (kg/m³) is the density of the sphere. Equating the two forces and introducing the values used in the simulation gives an approximate terminal velocity of 0.284 m/s.

The same reference discusses correction factors for non-spherical particles. You can easily adapt the model to hold for a general axially symmetric object (by redrawing the geometry) or even an arbitrarily shaped object (by modeling in 3D).

Reference

1. J.M. Coulson and J.F. Richardson, Chemical Engineering vol. 2, 4th ed., 1993.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/falling_sand

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select Axial symmetry (2D) from the Space dimension list.
- 2 In the list of application modes, select COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis.
- 3 Click Multiphysics, and then Add.
- 4 Click the Application Mode Properties button and set Weak constraints to On and Constraint type to Non-ideal.
- 5 Click **OK** to close the dialog box.

OPTIONS AND SETTINGS

I Enter the following constant names and expressions in the **Constants** dialog box:

NAME	EXPRESSION
rho_water	1000
eta_water	1.51e-3
r_grain	1e-3
V_grain	4/3*pi*r_grain^3
rho_grain	2900
m_grain	V_grain*rho_grain
g	9.81
Fg	-m_grain*g

2 Click **OK** to close the dialog box.

GEOMETRY MODELING

I Go to **Draw>Specify Objects** and specify a rectangle with the following properties:

PROPERTY	EXPRESSION
Width	6e-3
Height	14e-3
Position: Base	Corner
Position: r	0
Position: z	-6e-3

- 2 Click the **Zoom Extents** button to see the rectangle you just created.
- **3** Specify a circle centered at (0, 0) with the radius 10^{-3} .
- 4 Select both objects and click the **Difference** button.

PHYSICS SETTINGS

Space-Independent Equations

- I Choose Physics>Space-Independent Equations to open the Space-Independent Equations dialog box.
- **2** Enter the following ODEs:.

NAME	EQUATION	INIT (U)	INIT (UT)
Х	Xt-Xdot	0	0
Xdot	Xdott-(Fz+Fg)/m_grain	0	0

Boundary Conditions.

I In the **Boundary Settings** dialog box, enter boundary coefficients as indicated in the table below:

SETTINGS	BOUNDARIES 1, 3, 5	BOUNDARY 2	BOUNDARY 4	BOUNDARIES 6, 7
Туре	Symmetry boundary	Inlet	Open boundary	Wall
Condition	Symmetry	Velocity	Normal stress	No slip
v ₀		-Xdot		
f_0			0	

2 Choose Options>Integration Coupling Variables>Boundary Variables.

3 On the Source tab, select Boundaries 6 and 7. Specify a boundary integration coupling variable with the Name Fz and the Expression -2*pi*1m2. Leave the Integration order at the default value 4 and the Global destination set to on. The

Lagrange multiplier lm2 is equal to the *z* component of the total stress tensor times the *r* coordinate. Using the expression $-T_z_ns*2*pi*r$ would give a similar solution but with a worse accuracy and a slightly higher memory consumption.

4 Click **OK** to close the dialog box.

Subdomain Settings

In the **Subdomain Settings** dialog box, enter the subdomain settings indicated in the table below.

SETTINGS	SUBDOMAIN I
ρ	rho_water
η	eta_water
F _r	0
Fz	-rho_water*(Xdott+g)

MESH GENERATION

- I Open the Free Mesh Parameters dialog box
- 2 Select Fine among the Predefined mesh sizes.
- 3 Click the Custom mesh size button and Set the Element growth rate to 1.1.
- 4 On the **Boundary** tab, set the **Maximum element size** for Boundary 3 to 2e-4, and for Boundaries 6 and 7 to 1e-4.
- **5** Click the **Remesh** button.
- 6 Click **OK** to close the dialog box.

COMPUTING THE SOLUTION

- I Open the Solver Manager.
- 2 On the **Output** tab, select the **Include time derivatives** check box.
- **3** Click **OK** to close the dialog box.
- I Open the Solver Parameters dialog box.
- 2 On the General tab, select the Time dependent solver.
- 3 In the Times text field, enter 0:0.025:1.
- 4 In the Absolute tolerance text field, enter 1e-5.
- **5** Click **OK** to close the dialog box.
- **6** Click the **Solve** button.

POSTPROCESSING AND VISUALIZATION

The default plot shows the velocity field in the reference system of the grain.

- I Open the Plot Parameters dialog box.
- 2 On the Streamline tab, click to select the Streamline plot check box.
- 3 Click the Color button on the Line Color tab and select a black color.
- 4 Click **OK** to see the streamline plot.

You can also visualize the velocity and the forces on the grain as functions of time:

- 5 Choose Postprocessing>Domain Plot Parameters.
- 6 On the **Point** tab, select an arbitrary point and enter -Xdot in the **Expression** text field.
- 7 Click **Apply** to plot the velocity against time.

It is possible to view all the forces in the same figure:

- 8 On the Point tab, select an arbitrary point and enter Fz in the Expression text field.
- 9 Click Apply to plot the viscous and pressure contributions to the force against time.

10 On the General tab, click to select the Keep current plot check box.

II On the Point tab, click the Line Settings button and select a red color.

- **12** In the **Expression** text field, enter Fg.
- **I3** Click **Apply**.
- 14 Click the Line settings button again and select a black color.
- **I5** In the **Expression** text field, enter Fg+Fz.
- **I6** Click **OK** to reproduce the plot in Figure 6-14 and close the dialog box.

Sloshing Tank

Introduction

This 2D model demonstrates the ability of COMSOL Multiphysics to simulate dynamic free surface flow with the help of a moving mesh. The study models fluid motion with the incompressible Navier-Stokes equations. The fluid is initially at rest in a rectangular tank. The motion is driven by the gravity vector swinging back and forth, pointing up to 4 degrees away from the downward *y* direction at its extremes.



Figure 6-15: Snapshots of the velocity field at t = 1 s, t = 1.2 s, t = 1.4 s, and t = 1.6 s. The inclination of the gravity vector is indicated by the leaning of the tank.

Because the surface of the fluid is free to move, this model is a nonstandard computational task. The ALE (arbitrary Lagrangian-Eulerian) technique is, however, well suited for addressing such problems. Not only is it easy to set up using the Moving Mesh (ALE) application mode in COMSOL Multiphysics, but it also has the advantage that it represents the free surface boundary with a domain boundary on the

moving mesh. This allows for the accurate evaluation of surface properties such as curvature, making surface tension analysis possible. Note, however, that this example model neglects surface tension effects.

Model Definition

DOMAIN EQUATIONS

This model describes the fluid dynamics with the incompressible Navier-Stokes equations:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot (-p \mathbf{I} + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) = \mathbf{F}$$
$$\nabla \cdot \mathbf{u} = 0$$

where ρ is the density, $\mathbf{u} = (u, v)$ is the fluid velocity, p is the pressure, \mathbf{I} is the unit diagonal matrix, η is the viscosity, and \mathbf{F} is the volume force. In this example model, the material properties are for glycerol: $\eta = 1.49$ Pa·s, and $\rho = 1.27 \cdot 10^3$ kg/m³. The gravity vector enters the force term as

$$F_x = \rho g \sin(\phi_{\max} \sin(2\pi f t))$$
$$F_y = -\rho g \cos(\phi_{\max} \sin(2\pi f t))$$

where $g = 9.81 \text{ m/s}^2$, $\phi_{\text{max}} = 4\pi/180$, and f = 1 Hz.

With the help of the Moving Mesh (ALE) application mode, you can solve these equations on a freely moving deformed mesh, which constitutes the fluid domain. The deformation of this mesh relative to the initial shape of the domain is computed using Winslow smoothing. For more information, please refer to "The Moving Mesh Application Mode" on page 401 in the *COMSOL Multiphysics Modeling Guide*. COMSOL Multiphysics takes care of the transformation of the Navier-Stokes equations to the formulation on the moving mesh.

BOUNDARY CONDITIONS FOR THE FLUID

There are two types of boundaries in the model domain. Three solid walls, that are modeled with slip conditions, and one free boundary (the top boundary). The slip boundary condition for the Navier-Stokes equations is

 $\mathbf{u} \cdot \mathbf{n} = 0$

where $\mathbf{n} = (n_x, n_y)^T$ is the boundary normal. To enforce this boundary condition, select the Symmetry boundary type in the Incompressible Navier-Stokes application mode. Because the normal vector depends on the degrees of freedom for the moving mesh, a constraint force would act not only on the fluid equations but also on the moving mesh equations. This effect would not be correct, and one remedy is to use non-ideal weak constraints. Ideal weak constraints (the other type of weak constraints) do not remove this effect of the constraint force. For more information about weak constraints, see "Using Weak Constraints" on page 300 in the *COMSOL Multiphysics Modeling Guide*. The Incompressible Navier-Stokes application mode does not make use of weak constraints by default, so you need to activate the non-ideal weak constraints.

The following weak expression, which you add to the model, enforces the slip boundary condition without a constraint force acting on the moving mesh equations:

$$\hat{\lambda}(\mathbf{u} \cdot \mathbf{n}) - \lambda(\mathbf{u} \cdot \mathbf{n}) \tag{6-9}$$

for some Lagrange multiplier variable λ . Here λ and **u** denote test functions. See the step-by-step instructions later in this model documentation for details.

The fluid is free to move on the top boundary. The stress in the surrounding environment is neglected. Therefore the stress continuity condition on the free boundary reads

$$(-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) \cdot \mathbf{n} = -p_0\mathbf{n}$$

where p_0 is the surrounding (constant) pressure and η the viscosity in the fluid. Without loss of generality, $p_0 = 0$ for this model.

BOUNDARY CONDITIONS FOR THE MESH

In order to follow the motion of the fluid with the moving mesh, it is necessary to (at least) couple the mesh motion to the fluid motion normal to the surface. It turns out that for this type of free surface motion, it is important to not couple the mesh motion to the fluid motion in the tangential direction. If you would do so, the mesh soon becomes so deformed that the solution no longer converges. The boundary condition for the mesh equations on the free surface is therefore

$$(x_t, y_t)^T \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}$$

where **n** is the boundary normal and $(x_t, y_t)^T$ the velocity of mesh (see "Mathematical Description of the Mesh Movement" on page 392 in the *COMSOL Multiphysics*

Modeling Guide). In the Moving Mesh (ALE) application mode, you specify this boundary condition by selecting the tangent and normal coordinate system in the deformed mesh and by specifying a mesh velocity in the normal direction, where you enter the right-hand side expression from above as u*nx+v*ny. The Moving Mesh (ALE) application mode uses non-ideal weak constraints by default, and for this boundary condition it adds the weak expression

$$\hat{\lambda}(((x_t, y_t)^T - \mathbf{u}) \cdot \mathbf{n}) - \lambda(\hat{(x, y)}^T \cdot \mathbf{n})$$

to ensure that there are no constraint forces acting on the fluid equations. Here again, λ denotes some Lagrange multiplier variable (not the same as before) and λ , x, and y denote test functions. There is no need to modify this expression. Choose **Physics>Equation System>Boundary Settings** and select the free boundary (boundary 3) to see how to enter this expression in COMSOL Multiphysics. The expression implies that there is a flux (or force) on the free boundary for the moving mesh coordinate equations $\nabla x \cdot \mathbf{n} = \lambda n_x$ and $\nabla y \cdot \mathbf{n} = \lambda n_y$, respectively. Furthermore, to be able to follow the fluid motion with the mesh motion, the moving mesh must not be constrained in the tangential direction on the side walls. In the Moving Mesh (ALE) application mode, you specify this boundary condition by using the global coordinate system and setting the mesh displacement to zero in the x direction. At the bottom of the tank the mesh is fixed, which you obtain in a similar way by setting the mesh displacements to zero in both the x and y directions.

Results

Figure 6-15 on page 247 shows the tank at a few different points in time. The colors represent the velocity field. Whereas the modeling is set up using a fixed tank and a swinging gravity vector, postprocessing using a deformation plot gives the tank a corresponding inclination. The inclination angle of the tank is exactly the same as the angle of the gravity vector from its initial vertical position.

To illustrate the dynamics in the tank, you can plot the wave height versus time at one of the vertical walls, as in the following plot.



Figure 6-16: Wave height at X = 0.5 m for $0 \le t \le 20 s$.

The movie file that accompanies this model shows the waves in the swinging tank, with a color scale indicating the vorticity.

Model Library path: COMSOL_Multiphysics/Fluid_Dynamics/sloshing_tank

Modeling Using the Graphical User Interface

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator, click the Multiphysics button.
- 3 Select 2D from the Space dimension list.
- 4 Select COMSOL Multiphysics>Deformed Mesh>Moving Mesh (ALE)>Transient analysis and click Add.
- 5 Click the Application Mode Properties button.
- 6 Select Winslow from the Smoothing method list. Click OK.
- 7 Select COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis and click Add.

8 Click OK.

GEOMETRY MODELING

I Shift-click the **Rectangle/Square** button in the Draw toolbar.

2 Specify the rectangle settings according to the table below

PROPERTY	EXPRESSION
Width	1
Height	0.3
Position: Base	Corner
Position: x	-0.5
Position: y	0

3 Click the Zoom Extents button on the Main toolbar.

OPTIONS AND SETTINGS

I Open the **Constants** dialog box from the **Options** menu and enter the following constants. The descriptions are optional. When done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
rho	1270[kg/m^3]	Glycerol density
nu	1.49[Pa*s]	Glycerol viscosity
phi_max	(4*pi/180)[rad]	Maximum angle of inclination
freq	1[Hz]	Frequency
g	9.81[m/s^2]	Acceleration due to gravity

2 From the Options menu, choose Expressions>Scalar Expressions.

3 Enter the following scalar variables with names, expressions, and descriptions (the descriptions are optional); when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
phi	phi_max*sin(2*pi*freq*t)	Angle of inclination
grav_x	g*sin(phi)	Gravity vector x component
grav_y	-g*cos(phi)	Gravity vector y component

PHYSICS SETTINGS

Properties

- I In the Incompressible Navier-Stokes application mode, choose **Properties** from the **Physics** menu.
- **2** In the **Application Mode Properties** dialog box, select **On** from the **Weak constraints** list and **Non-ideal** from the **Constraint type** list; then click **OK**.

Subdomain Settings

Open the Subdomain Settings dialog box and apply the settings in the table below.

SETTINGS	SUBDOMAIN I
ρ	rho
η	nu
F _x	grav_x*rho
F _y	grav_y*rho

Boundary Conditions

I Open the **Boundary Settings** dialog box from the **Physics** menu and enter boundary conditions according to the table below.

SETTINGS	BOUNDARIES 1, 2, 4	BOUNDARY 3
Boundary type	Wall	Open boundary
Boundary condition	Slip	Normal stress
f ₀		0

2 Click OK.

- **3** Go to the Multiphysics menu and select Moving Mesh (ALE).
- **4** In the **Boundary Settings** dialog box, apply the following boundary conditions for the mesh displacements (only tangential movements on the sides and a fixed mesh at the bottom):

SETTINGS	BOUNDARIES I, 4	BOUNDARY 2
dx	0	0
dy		0

5 On Boundary 3, select Tangent and normal coord. sys. in deformed mesh in the Coordinate system list. Then click the Mesh velocity button and type u*nx+v*ny in the vn edit field to specify the normal mesh velocity as u·n.

- 6 On the Weak Constr. tab of the Boundary Settings dialog box, clear the Use weak constraints check box on Boundaries 1, 2, and 4. The strong constraints that you specified in the previous step are sufficient on these boundaries. Leave the Use weak constraints check box selected on Boundary 3.
- 7 Click **OK** to close the dialog box.

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar to initialize the mesh.

COMPUTING THE SOLUTION

- I Open the Solver Parameters window from the Solve menu.
- 2 Select Time dependent from the Solver list.
- 3 Enter 0:0.1:6 in the Times edit field.
- **4** Type 0.001 in the **Relative tolerance** edit field. This provides a 0.1% relative tolerance, which is one order of magnitude less than the default value.
- 5 Click the Time Stepping tab.
- **6** Select **Exclude algebraic** in the **Error estimation strategy** list. This excludes the pressure and the moving mesh variables from the error estimation. The equations for those variables do not include time derivatives and become algebraic when solving the equation system using the method of lines.
- 7 Click OK.
- 8 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The default plot shows the *x*-component of the moving mesh deformation, in the spatial frame.

- I To plot the velocity field of the glycerol instead, go to the Surface tab in the Plot Parameters dialog box and select Incompressible Navier-Stokes (ns)>Velocity field from the list of expressions.
- 2 On the **General** page, clear the **Geometry edges** check box. Click **Apply** to see the plot and use the **Solution at time** list on the **General** tab to browse through the output times.

It is possible to visualize the inclination of the tank by clever use of the deformation plot feature:

- 3 On the **Deform** page, select the **Deformed shape plot** check box and Set the **Scale** factor to 1. Enter Y*sin(phi) in the X component edit field and -X*sin(phi) in the Y component edit field on the Subdomain Data tab.
- 4 Still on the **Deform** page, click the **Boundary Data** tab. Once again, enter Y*sin(phi) in the X component edit field and -X*sin(phi) in the Y component edit field.
- **5** On the **Boundary** tab, select the **Boundary plot** check box. Enter **1** in the **Expression** edit field. Select to use a **Uniform color** and pick a black color using the **Color** button.
- 6 To get a more liquid-looking plot, you may want to go to the **Surface** page and set the **Colormap** to **bone**.
- 7 Click **Apply** to see the plot.
- 8 To see the waves in action, go to the Animate tab and click Start Animation, then click OK.

To get a more comprehensive overview of the sloshing, you can plot the *y*-displacement from equilibrium in a point:

- I Open the Domain Plot Parameters dialog box from the Postprocessing menu.
- 2 On the **Point** tab, select Point 4 from the **Point selection** list.
- 3 Enter dy_ale in the Expression edit field, then click OK to see the plot.

Geophysics Models

7

This chapter presents models of geophysics applications such as groundwater flow, contaminant transport, and the flow of fluids in rock fractures. You will find many more models and specialized application modes in the *Earth Science Module*.

Groundwater Flow and Solute Transport

Introduction

This model demonstrates the application of COMSOL Multiphysics to a benchmark case of steady-state subsurface fluid flow and transient solute transport along a vertical cross section in an unconfined aquifer. Because of profound geologic heterogeneity, the model must estimate solute transport subject to highly irregular flow conditions with strong anisotropic dispersion. Van der Heijde (Ref. 1) classifies this case as "Level 2," with enough potentially difficult parameter combinations to test a code's ability to tackle realistic hydrologic situations. Sudicky (Ref. 2) developed this problem to demonstrate a Laplace transform Galerkin code. This problem subsequently has been used to evaluate other flow and transport models, including MT3DMS by Zheng and Wang (Ref. 3).

This model makes use of several useful COMSOL Multiphysics features:

- · Multiphysics coupling between fluid flow and solute transport
- · Freely defined equations using the PDE, Coefficient Form application mode
- Upper boundary is water table with time-dependent solute source
- · Incorporation of subsurface geological heterogeneity
- · Customized expressions used to create anisotropic dispersion tensor

Model Definition

The hydrologic setting for this problem is described in Figure 7-1 (a) (see Ref. 2), for groundwater flow at steady state. The aquifer is composed largely of fine-grained silty sand of hydraulic conductivity $K_1 = 5 \cdot 10^{-4}$ cm/s, equivalent to $K_1 = 5 \cdot 10^{-6}$ m/s, with lenses of relatively course material of hydraulic conductivity $K_2 = 1 \cdot 10^{-2}$ cm/s, equivalent to $K_2 = 1 \cdot 10^{-4}$ m/s. Generally, groundwater moves from the upper surface of the saturated zone, the water table, to the outlet at x = 250 m. The water table is a free surface, that is, fluid pressure equals zero, across which there is vertical recharge, denoted R, of 10 cm/yr equivalent to $3.215 \cdot 10^{-9}$ m/s. The groundwater divide, a line of symmetry, occurs at x = 0. The base of the aquifer is impermeable.

Figure 7-1 (b) shows conditions related to solute transport. The aquifer initially is pristine, and concentrations equal zero. For the first five years, a relative concentration of 1.0 is loaded over the interval 40 m < x < 80 m at the water table. The solute source is removed in year 5, and the concentration along this segment immediately drops to zero. The contaminant migrates within the aquifer via advection and dispersion. Throughout the domain, porosity, denoted *n*, is 0.35, the longitudinal dispersivity, α_L , and transverse vertical dispersivity, α_T , are 0.5 m and 0.005 m, respectively, and the effective molecular diffusion coefficient, D_m , is $1.34 \cdot 10^{-5}$ cm²/s, equivalent to $1.34 \cdot 10^{-9}$ m²/s.



Figure 7-1: Definition of the flow-field problem (a) and the transport-of-solute problem (b).

FLUID FLOW: DOMAIN EQUATIONS AND BOUNDARY CONDITIONS

Governing equations for fluid flow and solute transport were specified in separate PDE, Coefficient Form application modes in COMSOL Multiphysics.

Steady groundwater flow generally is expressed with a conservation equation built with Darcy's law (Ref. 4, Ref. 5):

$\nabla \cdot (K \nabla h) + R = 0$

where *K* is hydraulic conductivity (L/T); x_i is spatial distance in direction *i* (L); *R* is the volumetric rate of recharge to water table per unit volume of aquifer (T⁻¹), and the dependent variable *h* is hydraulic head (L). Hydraulic head, a function of pressure and gravitational potential, is defined

$$h = h_p + y \tag{7-1}$$

where h_p is pressure head (L); and y is elevation (L). Equation 7-1 states the driving force for groundwater flow at field scales is h. For any given water particle, h equals the height of the water column h_p above the particle plus the particle's elevation y. It should be pointed out here that y is the name assigned to the independent variable x_i for the vertical direction.

The equations for groundwater flow and solute transport are linked by the average linear velocity v, or seepage velocity:

$$v_i = -\frac{K\partial h}{\partial x_i} \tag{7-2}$$

where *n* is porosity (L^3/L^3) , or the fraction of the aquifer containing water. *n* appears in the denominator of Equation 7-2 because only a portion of a given aquifer block is available for flow.

The boundary conditions for the groundwater flow problem are shown in Figure 7-1 (a) and stated below. A zero flux Neumann condition represents the symmetry boundary at x = 0 m and the impermeable boundary at y = 0 m as follows:

$$\left. \frac{\partial h}{\partial x} \right|_{x = 0} = \left. \frac{\partial h}{\partial y} \right|_{y = 0} = 0$$

Hydraulic head is specified at x = 250 m with a Dirichlet condition:

$$h(x,t) = h_0$$

Representing the water table is slightly more complicated. A Neumann boundary is used to model the known recharge:

$$-K\frac{\partial h}{\partial y} = R$$

Specifying that this flux is entirely vertical requires multiplication by the y component of the normal vector **n** in COMSOL Multiphysics.

An important advantage of specifying a flux condition on the water table is the opportunity to fine tune the model setup. If the flow problem is well posed, h naturally equals y at the water table, because h_p is zero (that is, atmospheric pressure) at a free surface. In this model, the water table geometry is first determined by educated guess and is fine tuned manually in successive simulations until there is a good match between elevations and hydraulic head predictions on the boundary (Ref. 2).

SOLUTE TRANSPORT: DOMAIN EQUATIONS, BOUNDARY CONDITIONS, AND INITIAL CONDITIONS

Solute transport typically is time dependent for geologic problems and is described with the advection-dispersion equation (Ref. 4, Ref. 5):

$$\frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial C}{\partial x_j} + v_i C \right) = \frac{\partial C}{\partial t}$$

where D_{ij} is the hydrodynamic dispersion tensor (L²/T); *C* is the dissolved concentration (M/L³); v_i is the average linear velocity (defined above); and *t* is time.

The dispersion tensor defines solute spreading by mechanical mixing and molecular diffusion. Equations for the tensor entries are:

$$D_{ii} = \alpha_L \frac{v_i^2}{|v|} + \alpha_T \frac{v_j^2}{|v|} + D^*$$
$$D_{ij} = D_{ji} = (\alpha_L - \alpha_T) \frac{v_i v_j}{|v|} + D^*$$

where D_{ii} are the principal components of the dispersion tensor (L^2/T) ; D_{ij} , D_{ji} are the cross terms of the dispersion tensor (L^2/T) ; the subscript L denotes longitudinal dispersivity (L); the subscript T denotes transverse dispersivity (L); $|v| = \sqrt{v_x^2 + v_y^2}$ is the magnitude of the velocity vector (L/T); and D^* represents effective molecular diffusion D_m in saturated porous media, (L^2/T) , where $D^* < D_m << D_{ii}$ and typically is neglected.

The boundary and initial conditions for solute transport, shown in Figure 7-1 (b), are expressed below. Dirichlet conditions are used at the water table, where C(x,h,t) = 0, except for the segment 40 m < x < 80 m in which

$$C = \begin{cases} C_0, & 0 < t < t_0 \\ 0, & t > t_0 \end{cases}$$

where the relative concentration C_0 is 1.0 through year 5 (t_0). After the solute source is removed, the concentration along this segment drops to zero. Here, the time dependence of the source is implemented using logical expressions with COMSOL Multiphysics boundary settings.

The Dirichlet condition at the left boundary is

$$C(0, y, t) = 0$$

A Neumann condition is needed for the zero gradient boundaries:

$$\left. \frac{\partial h}{\partial x} \right|_{x = 250} = \left. \frac{\partial h}{\partial y} \right|_{y = 0} = 0$$

Finally, the initial condition specifies that the aquifer is pristine, when

$$C(x, y, 0) = 0$$

Results

FLUID FLOW

Figure 7-2 provides hydraulic heads estimated with the COMSOL Multiphysics steady-state groundwater flow simulation. The hydraulic heads and streamlines shown in Figure 7-2 correspond nicely to the benchmark results provided by Sudicky (Ref. 2). The slight differences between the two plots are attributable to meshing: the mesh size varies naturally with the geometry.

The water table geometry determined for the simulation (see Figure 7-2) nearly duplicates the benchmark geometry obtained in Ref. 2. The good match between the flow fields is expected as the initial water table geometry used with COMSOL Multiphysics was designed to closely resemble the benchmark geometry. In the simulations reported here, therefore, only nuances of geometry were resolved through iteration. Figure 7-3 provides residuals from the geometry fitting; that is, the fractional

difference between h and y along the final water table boundary. As Figure 7-3 illustrates, the error in the geometry settings are 0.1% or less.



Figure 7-2: COMSOL Multiphysics estimates of hydraulic head and flow lines. The solution reproduces the results in Ref. 3.



Figure 7-3: Fractional error in water table geometry computed with hydraulic head predicted with COMSOL Multiphysics, H, and water table elevation, y.

SOLUTE TRANSPORT

Solute transport solutions from the model are almost identical to the ones presented in Ref. 2. This is clearly shown in the contour intervals for three times in Figure 7-4. The minor variation between the COMSOL Multiphysics and benchmark solutions results from different mesh densities. In 1989, Sudicky concluded that the results illustrated in Ref. 2 are relatively free of numerical dispersion, as the low concentration contours closely follow the flow pattern. The surface plot for COMSOL also displays this property, in that even the lowest concentrations in Figure 7-4 still follow the irregular flow lines of Figure 7-2.



Figure 7-4: Plume concentrations calculated with COMSOL Multiphysics at three times: (a) t = 8 years, (b) t = 12 years, and (c) t = 20 years.

The results from this model show that COMSOL Multiphysics is an effective tool for simulating fluid flow and solute transport in the heterogeneous porous media and highly irregular flow fields that are common to field scale applications.

References

1. P.K.M. van der Heijde, "Model Testing: A Functionality Analysis, Performance Evaluation, and Applicability Assessment Protocol," *Groundwater Models for Resources Analysis and Management*, A.I. El-Kadi (ed.), CRC Press, Lewis Publishers, Boca Raton, FL, pp. 39–58, 1995.

2. E.A. Sudicky, "The Laplace transform Galerkin technique: application to mass transport in groundwater," *Water Resour. Res.*, vol. 25, no. 8, pp. 1833–1846, 1989.

3. C. Zheng and P. Wang, *MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion and Chemical Reactions of Contaminants in Groundwater Systems*, University of Alabama, 239 pp, 1998.

4. J. Bear, *Dynamics of Fluids in Porous Media*, Elsevier Scientific Publishing Company, 764 pp, 1972.

5. J. Bear, Hydraulics of Groundwater, McGraw-Hill, New York, 210 pp, 1979.

Model Library path: COMSOL_Multiphysics/Geophysics/groundwater_flow

Modeling Using the Graphical User Interface

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator click the Multiphysics button.
- **3** Select COMSOL Multiphysics>PDE Modes>PDE, Coefficient Form and type H in the Dependent variables edit field.
- 4 Click the Add button.
- 5 Select PDE, Coefficient Form again.
- 6 Change the name of the dependent variable to C in the Dependent variables edit field.
- 7 Click the Add button.
- 8 Click OK.

OPTIONS AND SETTINGS

- I Choose Axes/Grid Settings from the Options menu.
- 2 Clear the Axis equal check box.
- **3** Set the *x*-*y* limits according to following table:

LIMIT	VALUES
x min	0
x max	270
y min	-2
y max	10

4 Click the Grid tab.

- 5 Clear the Auto check box.
- 6 Type 10 in the x spacing edit field and 2 in the y spacing edit field.
- 7 Add extra grid lines by typing 155 127 in the Extra x edit field and 5.35 5.575
 6.045 6.455 6.603 6.645 in the Extra y edit field. Click OK.
- 8 Choose Constants from the Options menu and define the following constants:

NAME	EXPRESSION
R	3.215e-9
Hin	5.3486
K1	5e-6
К2	0.0001
Cin	1
n	0.35
alphaL	0.5
alphaT	0.005
Dm	1.34e-9

GEOMETRY MODELING

- Click the Line button to draw the geometry of the domain by clicking in the points (0, 0), (250, 0), (250, 5.35), (180, 5.575), (155, 6.045), (127, 6.455), (80, 6.603), (0, 6.645).
- **2** Click the **Rectangle/Square** button and click the points (0, 2) and (120, 4) to draw a rectangle.
- **3** Draw a second rectangle with corners in the points (180, 2) and (250, 4).

PHYSICS SETTINGS

Subdomain Expressions

I On the **Options** menu, point to **Expressions** and then click **Subdomain Expressions**.

2 Enter the following expressions in the Subdomain Expressions dialog box:

NAME	EXPRESSION IN SUBDOMAIN I	EXPRESSION IN SUBDOMAIN 2 AND 3
VX	- K1 *Hx / n	-K2*Hx/n
vy	- K1 *Hy / n	-K2*Hy/n
absv	sqrt(vx^2+vy^2)	<pre>sqrt(vx^2+vy^2)</pre>
Dxx	alphaL*vx^2/ absv+alphaT*vy^2/absv+Dm	alphaL*vx^2/ absv+alphaT*vy^2/absv+Dm
Dyy	alphaL*vy^2/ absv+alphaT*vx^2/absv+Dm	alphaL*vy^2/ absv+alphaT*vx^2/absv+Dm
Dxy	(alphaL-alphaT)*vx*vy/absv	(alphaL-alphaT)*vx*vy/absv

3 Click OK.

Boundary Conditions—Darcy's Law

- I Choose I PDE, Coefficient Form (c) from the Multiphysics menu.
- **2** Open the **Boundary Settings** dialog box and enter the boundary condition according to the following table:

SETTINGS	BOUNDARIES1-3, 5	BOUNDARIES 7, 8, 10, 11, 15	BOUNDARIES 16-18
Туре	Neumann	Neumann	Dirichlet
q	0	0	
g	0	-ny*R	
h			1
r			Hin

Subdomain Settings—Darcy's Law

- I Select Subdomain 1 and set c equal to -K1. All the other coefficients should be 0.
- **2** Select Subdomains 2 and 3.
- **3** Set **c** equal to -K2. All the other coefficients should be 0.

Boundary Conditions—Advection-Dispersion Equation

I Choose 2 PDE, Coefficient Form (c2) from the Multiphysics menu.

SETTINGS	BOUNDARIES 1, 3, 5, 8, 10, 11, 15	BOUNDARY 2	BOUNDARY 7	BOUNDARIES	BOUNDARY 17
Туре	Dirichlet	Neumann	Dirichlet	Neumann	Neumann
q		0		0	0
g		0		K1*Hx/n*C	K2*Hx/n*C
h	1		1		
r	0		Cin*(40<=x&t<= 5*360*86400)		

2 Enter the boundary conditions according to the following table:

Subdomain Settings—Advection-Dispersion Equation

Select all domains and enter the coefficients according to the following table:

COEFFICIENT	VALUE
c	Dxx Dxy Dyy
а	0
f	0
da	1
α	-vx -vy
β	0 0
γ	0 0

MESH GENERATION

Click the **Refine Mesh** button on the Main toolbar to initialize an unstructured triangular mesh and then refine it once.

COMPUTING THE SOLUTION

- I Open the Solver Manager dialog box and click the Solve For tab.
- 2 Select PDE, Coefficient Form (c), then click Solve.
- **3** Select **PDE**, **Coefficient Form (c2)**, then click **OK**.
- 4 Open the Solver Parameters dialog box.
- 5 Select Time dependent in the Solver list.
- 6 Type 0:360*86400:20*360*86400 in the Times edit field. Click OK.
- 7 Click the **Restart** button to compute a new solution.

POSTPROCESSING AND VISUALIZATION

Use the first sequence of commands to plot Figure 7-2 on page 263:

- I From the Postprocessing menu, choose Plot Parameters.
- 2 Click the **Surface** tab.
- **3** On the Surface Data tab, select PDE, Coefficient Form (c)>H from the Predefined quantities list.
- 4 Click the Streamline tab.
- **5** Select the **Streamline plot** check box.
- 6 On the Streamline Data tab, select PDE, Coefficient Form (c)>grad(H) from the Predefined quantities list.
- 7 Click the Specify start point coordinates button and enter the following values

x 15 30 45 60 75 90 130 145 160 175 190 205 220 235 y 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

- 8 Click the Advanced button and enter 40000 in the Maximum number of integration steps edit field. Click OK.
- 9 Click OK.

You can also plot the plum concentrations in Figure 7-4 on page 264:

- I Open Plot Parameters dialog box again.
- 2 Click the Surface tab and select PDE, Coefficient Form (c2)>C from the Predefined quantities list on the Surface Data tab.
- **3** Click the **Contour** tab and select **PDE**, **Coefficient Form** (c2)>C from the **Predefined quantities** list on the **Contour Data** tab.
- In the Contour levels area, click the Vector with isolevels button and enter 0.05 0.1
 0.15 0.19 0.25 0.30 0.35 0.4 0.45 0.5 in the edit field.
- 5 Click the **General** tab, select the **Contour** check box, and clear the **Streamline** check box. Click **OK**.
- **6** To get a good view of the solution, double-click **EQUAL** on the status bar and then click the **Zoom Extents** button on the Main toolbar.

To plot this concentration for a different time you can select the time you want in the **Solution at time** list on the **General** page in the **Plot Parameters** dialog box (for Figure 7-4 (a) select **2.48832e8**, for Figure 7-4 (b) select **3.73248e8**, and for Figure 7-4 (c) select **6.2208e8**, that is, the concentration after 8, 12, and 20 years).

A Rock Fracture Flow Model

Introduction

A potential flow model describing fluid movement in a rock fracture uses the so-called Reynolds equation, also known as the "cubic law" equation

$$\nabla \cdot \left(\frac{\rho g}{12\mu}a^3\nabla h\right) = 0$$

which involves the following variables:

- Fluid density, ρ
- The gravitational constant, g
- The fluid's dynamic viscosity, μ
- The fracture's aperture or width, a(x, y)
- The scaled pressure, h = h(x, y), also called hydraulic head

This model uses interpolation of aperture data defined in a text file. Alternatively, you can use COMSOL Multiphysics' ability to call M-file functions. To do so, you need to use COMSOL Multiphysics with COMSOL Script or MATLAB.

Model Definition

The definition of that last variable, hydraulic head is

$$h = z + \frac{p}{\rho g}$$

where z equals the height and p represents fluid pressure.

After eliminating the constant factor in the diffusion coefficient, you end up with the equation

$$\nabla \cdot (a^3 \nabla h) = 0$$

Notice that you cannot eliminate the aperture expression a^3 because this example assumes that a is a nonconstant function of x and y, and thus it falls under the influence of the divergence operator.

Modeling in COMSOL Multiphysics

COMSOL Multiphysics does not include an application mode for potential flow, but you can use the Diffusion application mode for this model; the PDE solved here is identical to the Reynolds equation. You must make one minor mental adjustment here: In the user interface you refer to hydraulic head as a concentration (c).

The computational domain is rectangular and well inside the sampled aperture data matrix. Set an hydraulic head of 20 mm at the upper boundary, and 0 mm at the lower boundary. This creates a pressure difference of 20 mm that drives the fluid flow. Both the left and right boundaries have insulation/symmetry boundary conditions.

INTERPOLATING USING THE APERTURE DATA

The COMSOL CD provides two files containing the sample aperture data for this model. This synthetically generated data set corresponds to an aperture with a fractal dimension of 2.6. The 100-by-100 matrix of data is available as a text file, aperture_data.txt, for use with the COMSOL Multiphysics interpolation feature and as a MAT-file, flaperture.mat. If you use COMSOL Multiphysics with COMSOL Script or MATLAB, you can load the aperture data from the workspace using the command

load flaperture

To plot the aperture data, type

```
surf(aperture)
```

The surf command uses bilinear interpolation for visualizing the data. To rotate the plot from COMSOL Script, just click and drag in the COMSOL figure window. In MATLAB, you can rotate the 3D surface plot by pressing the **Rotate 3D** toolbar button in the plot window.

To interpolate the rectangular grid data to the nodes of the unstructured mesh, use the function flafun. To display that function, type the following at command prompt:

type flafun

Next note that the function interp2 uses linear interpolation as its default method. For more information on this function, type help interp2 at the command line.

When modeling this problem in the COMSOL Multiphysics user interface, the software uses flafun (either defined directly as an interpolation function or using the M-file) as the aperture a in the cubic-law equation.

Model Library path: COMSOL_Multiphysics/Geophysics/rock_fracture

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select 2D in the Space dimension list.
- 2 In the COMSOL Multiphysics>Convection and Diffusion folder, select Diffusion and then Steady-state analysis. Make sure Lagrange - Quadratic is selected in the Element list.
- 3 Click OK.

OPTIONS AND SETTINGS

Set axis and grid settings:

- I From the **Options** menu, choose **Axes/Grid Settings**.
- 2 Enter the axes limits and grid spacings from the table below. To enter the grid spacings, first click the **Grid** tab and click to clear the **Auto** check box.

AXIS		GRID	
x min	0	x spacing	10
x max	100	Extra x	
y min	0	y spacing	10
y max	100	Extra y	

3 Click OK.

Define the interpolation function for the aperture data:

- I From the **Options** menu, choose **Functions**.
- 2 In the Functions dialog box, click the New button.
- 3 In the New Function dialog box, type flafun in the Function name edit field.
- 4 Click Interpolation and select File in the Use data from list.
- 5 Click Browse and browse to /models/COMSOL_Multiphysics/Geophysics/ aperture_data.txt in the COMSOL installation directory.
- 6 Click **Open** to select this file and close the **Open Data** dialog box and then click **OK** to close the **New Function** dialog box.

7 Click OK.

Alternatively, if you use COMSOL Multiphysics with COMSOL Script or MATLAB, define the function flafun:

- I Open the Functions dialog box from the **Options** menu.
- 2 Click New to define a new function
- **3** Type flafun in the Function name edit field and select Analytic in the New Function dialog box. Click OK.
- **4** Type x, y in the **Arguments** edit field and then type flafun(x,y) in the **Expression** edit field.
- 5 Click OK.

Defining the flafun function in this way is not strictly necessary because COMSOL Multiphysics assumes that an undefined function name is a COMSOL Script or MATLAB function. The derivative is set to zero if you do not define a function. In cases where a dependent variable is used as function argument, or if the function has complex output for real input, the function definition is necessary.

GEOMETRY MODELING

- I Draw a rectangle with corners at (10, 20) and (90, 70).
- **2** Click the **Zoom Extents** button on the main toolbar to adjust the coordinate system with respect to the size of the rectangle.

PHYSICS SETTINGS

Boundary Conditions

Enter boundary coefficients as below:

SETTINGS	BOUNDARIES I, 4	BOUNDARY 2	BOUNDARY 3
Туре	Insulation/Symmetry	Concentration	Concentration
c ₀		0	20

Subdomain Settings

Enter the following PDE coefficients:

SETTINGS	SUBDOMAIN I
D (isotropic)	flafun(x[1/m],y[1/m])^3
R	0

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box.
- 2 Select the Adaptive mesh refinement check box on the General page.
- **3** Click the **Adaptive** tab.
- **4** Type 10 in the **Maximum number of refinements** edit field and 10000 in the **Maximum number of elements** edit field. This number might be too high depending on the amount of memory in your system. Users uncertain of available memory should start with a lower number.
- **5** Click **OK** to close the dialog box.
- 6 Click the Solve button on the Main toolbar to compute the solution.

POSTPROCESSING AND VISUALIZATION

- I Open the Plot Parameters dialog box from the Postprocessing menu.
- 2 Click the Surface tab, and then on the Surface Data page, select Diffusion (di)>Diffusive flux, c from the Predefined quantities list.
- 3 Still on the Surface page, click the Height Data tab. Select the Height data check box and then select Diffusion (di)>Concentration, c from the Predefined quantities list.
- 4 On the **General** page, open the **Title** dialog box. Click the right button to specify the title manually and type hydraulic head.
- 5 Click OK to close the Title dialog box, and then click Apply to plot.


- 6 To view the aperture data, click the Surface tab and type flafun(x,y) in the Expression edit field on the Height Data tab.
- 7 On the **General** page, open the **Title** dialog box. Click the right button to specify the title manually and type aperture (flafun(x,y)).
- 8 Click **OK** twice.



Heat Transfer Models

The following models illustrate heat transfer using the Conduction application mode. Heat transfer also takes part in many multiphysics models. For more information about the heat transfer application modes and three introductory models, see "Heat Transfer" on page 163 in the *COMSOL Multiphysics Modeling Guide*.

Heating with a Moving Laser

Laser beams are commonly used to locally heat the surface of various substrates, for example, in laser welding or thermal annealing such as on layered silicon devices. The laser beam typically moves over a surface in a periodic fashion to produce the desired localized heating. In case of layered silicon devices, each layer is very thin, making the modeling of the penetration depth caused by the moving laser a strongly time-dependent problem.



Figure 8-1: A moving laser heats a thin silicon substrate.

This example models the localized transient heating caused by a laser beam that moves in circles over a silicon substrate. The beam's penetration depth, which you can describe with an absorption coefficient, depends on the ambient temperature. The geometry under study represents the top layer of a silicon device. The model examines the penetration depth and the influence of the laser motion on the transient temperature distribution.

This model considers the laser beam as having an infinitesimal width and thus treats it as a line heat source. As such it is not meaningful to study the maximum temperature because it is mesh dependent. However, the overall heat flux and temperature distribution on a macroscopic level are both accurate. The model simulates the substrate as a 3D object (Figure 8-1) with these dimensions:

- Thickness: 1 mm
- Width: 10 mm-by-10 mm

It handles the variation of laser intensity with penetration depth using a 1D geometry that representing the substrate's thickness.



Figure 8-2: The 1D model geometry.

The model makes use of the Conduction application mode to describe the transient heat transfer in the 3D geometry. The transient energy-balance equation for heat conduction is

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

where ρ is the density, C_p is the specific heat capacity, \underline{k} is the thermal conductivity, and Q is the heat source term, here set to zero (this case models the source in a different way).

The material properties are those of silicon, using an anisotropic conductivity of $(k_x, k_y, k_z) = (163, 163, 16)$ in units of W/(m·K), a density of 2330 kg/m³, and a specific heat capacity of 703 J/(kg·K).

For the model, assume the boundaries are insulating.

In the 1D geometry, this model uses the Weak Form, Subdomain application mode to model the laser penetration. In the equation describing the penetration

$$\frac{\partial I}{\partial x} = -k_{\rm abs}I$$

I represents the relative laser intensity (the variable in the Weak Form, Subdomain application mode) and k_{abs} is the absorption coefficient. The absorption coefficient can depend on the temperature, and the expression used in this model is

$$k_{\rm abs} = 8 \cdot 10^3 - 10(T - 300 \text{ K}) \text{ (m}^{-1})$$

The volumetric heat source term, Q, in the 3D geometry is then

$$Q = P_{in}k_{abs}I$$

where $P_{\rm in}$ is the total power of the incoming laser beam.

Both of these equations are included in the Weak Form, Subdomain application mode, where they become one equation:

```
I_test*(Ix-k_abs*I)+k_abs*I*P_in*T_test
```

The first part of this expression describes the penetration equation, and the second part comes from the heat-source term in the 3D Heat Transfer application mode.

At the left boundary the model applies a homogeneous Neumann condition, and at the right boundary it sets the relative intensity, I, to unity. The total incoming laser power, P_{in} , is 50 W.

The model implements the heat source's motion when coupling the 3D temperature variable, T, to the 1D equation. It does so with a subdomain extrusion coupling variable using a general transformation. A time-dependent transformation expression results in a moving heat source. This case describes a circular repeating motion using the transformation expressions

$$x = R\sin(\omega t), \quad y = R\cos(\omega t), \quad z = x^{2}$$

where *x*, *y*, and *z* correspond to the 3D coordinates, and *x*' represents the 1D coordinate. Furthermore, *R* is the radius of circular motion, ω is the angular velocity, and *t* is time. The model uses the parameter values *R* = 0.02 m and ω = 10 rad/s.

This method—using a separate geometry and equation to model the source term—is very useful because it provides that term directly at the test-function level. Furthermore, it models the source motion separately with the transformation expressions, making it simple to alter. It is indeed the best way to model a moving point or line source.

The 3D model makes use of an extruded triangular mesh, which has a fine resolution close to the laser incident line and is coarse elsewhere. This results in a high-resolution

solution with minimum computation requirements. The mesh results in around 10,000 elements and 6200 degrees of freedom.



Figure 8-3: The 3D mesh produced by extruding a 2D triangular mesh, refined along the circular laser incident line.

Results and Discussion



Figure 8-4 depicts the temperature distribution at the laser-beam incident surface.

Figure 8-4: Temperature distribution after 1 s of laser heating.

The figure clearly shows a hot spot where the laser beam is located at a specific time. Furthermore, the results show a cold side and a warm side next to the vertical line below the laser beam. The warm side represents the area where the beam has just swept through.



A better way to study these effects is by plotting the temperature at the top surface along the circular laser-beam incident pattern as in Figure 8-5.

Figure 8-5: Temperature distribution along the laser-beam incident trajectory on the top surface after 0.7 s.

Here the laser beam moves from right to left, and the warm side is on the right side of the peak. Locally the temperature reaches around 510 K, but this value is completely mesh dependent. Nevertheless, the temperature distribution just a few mesh elements away represents the real temperature quite well.



Finally, Figure 8-6 shows beam penetration into the substrate. The heating at the bottom of the substrate is practically zero.

Figure 8-6: Relative laser-beam intensity as a function of sample depth.

Model Library path: COMSOL_Multiphysics/Heat_Transfer/laser_heating

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select **3D** in the **Space dimension** list and click the **Multiphysics** button.
- 2 Select Heat Transfer>Conduction>Transient analysis in the list of application modes (in the COMSOL Multiphysics folder if the license contains additional modules).
- 3 Select Lagrange Linear in the Element list and click Add.
- 4 Click Add Geometry and select ID in the Space dimension list. Click OK to close the dialog box.

- 5 Select COMSOL Multiphysics>PDE Modes>Weak form, Subdomain>Time-dependent analysis in the list of application modes.
- 6 Enter I in the Dependent variables text field.
- 7 Click Add.
- 8 Click OK to exit the Model Navigator and begin modeling.

OPTIONS AND SETTINGS

In the **Constants** dialog box enter the following names and expressions. The descriptions are optional.

NAME	EXPRESSION	DESCRIPTION
P_in	50[W]	Laser beam power
r	2[cm]	Radius of trajectory circle
omega	10[rad/s]	Angular velocity

GEOMETRY MODELING 3D

- I Click the **Geom I** tab to go to the 3D geometry.
- 2 Open the Work-Plane Settings dialog box in the Draw menu. Click OK to get a default xy work plane at z = 0.
- **3** Select **Draw>Specify Objects>Square** to create a square with the width 0.1 m, centered at the origin (select **Center** from the **Base** list).
- **4** Select **Draw>Specify Objects>Circle** to create a circle with the radius 0.02 m, centered at the origin.
- 5 Click the **Zoom Extents** button on the Main toolbar.

MESH GENERATION 3D

- I Open the Mesh>Free Mesh Parameters dialog box.
- 2 Select Coarser from the Predefined mesh sizes list.
- 3 On the Boundary page select Boundaries 5–8 and enter 1e-3 in the Maximum element size text field.
- 4 Click **OK** to close the dialog box.
- 5 Click the Initialize Mesh button to generate the mesh.
- 6 Select Mesh>Extrude Mesh and enter 1e-3 in the Distance text field.
- 7 On the Mesh page enter 5 for the Number of element layers and click OK.

GEOMETRY MODELING ID

- I Click the **Geom2** tab to go to the 1D geometry.
- 2 Specify a line from 0 to 1e-3, then click the Zoom Extents button.

MESH GENERATION ID

- I Open the Free Mesh Parameters dialog box.
- 2 On the **Boundary** page select Boundary 2. Enter 1e-5 for the **Maximum element size** and 1.1 for the **Element growth rate**.
- 3 Click **OK** to close the dialog box.
- **4** Click the **Initialize Mesh** button to generate the mesh.

PHYSICS SETTINGS 3D

Subdomain Settings

- I Click the **GeomI** tab to go to the 3D geometry.
- I Open the Physics>Subdomain Settings dialog box and select both subdomains.
- 2 Click Load and select Silicon from the Materials/Coefficients Library. Click OK when done.
- **3** Select **k** (anisotropic) and enter 163, 163, and 16 for the diagonal elements **kxx**, **kyy**, and **kzz**, respectively.
- 4 On the **Init** page enter 300 for the initial temperature.
- **5** Click **OK** to close the dialog box.

Boundary Conditions

Use the default **Thermal insulation** boundary condition for all exterior boundaries.

Coupling Variables

- I Select Options>Extrusion Coupling Variables>Subdomain Variables.
- 2 Select both subdomains and define a variable with the Name T and the Expression T. Select to use a General transformation and make sure the source transformation reads x: x, y: y, z: z.
- 3 On the Destination page select Geometry: Geom2 and Level: Subdomain. Select the check box in front of Subdomain 1 and apply the destination transformation x: r*sin(omega*t), y: r*cos(omega*t), z: x.
- 4 Click **OK** to close the dialog box.

PHYSICS SETTINGS ID

Subdomain Settings

- I Click the Geom2 tab to go to the 1D geometry.
- 2 Open the Physics>Subdomain Settings dialog box and select Subdomain 1.
- 3 On the Weak page enter (Ix-I_abs)*I_test+I_abs*P_in*T_test in the weak text field and 0 in the dweak text field.
- 4 On the lnit page enter 1 for the lnitial value $l(t_0)$.
- 5 Click **OK** to close the dialog box.

Boundary Settings

- I Open the Physics>Boundary Settings dialog box,
- 2 Enter 0 in the **constr** text field for Boundary 1, then enter 1-I in the same text field for Boundary 2.
- 3 Click **OK** to close the dialog box.

Scalar Expressions

In the Scalar Expressions dialog box make the following entries; when done, click OK.

NAME	EXPRESSION	DESCRIPTION
k_abs	8e3[1/m]-10[1/(m*K)]*(T-300[K])	Absorption coefficient
I_abs	k_abs*I	Relative absorption

COMPUTING THE SOLUTION

I In the Solve>Solver Parameters dialog box make the following changes, then click OK.

FIELD	VALUE
Times	0:0.02:1
Absolute tolerance	1
Linear system solver	Direct (UMFPACK)
Matrix symmetry	Nonsymmetric

2 Click the Solve button on the Main menu to compute the solution.

POSTPROCESSING AND VISUALIZATION

I f you are still in the 1D geometry, click the GeomI tab and then the Postprocessing Mode button to go over to postprocessing in 3D.

The default plot shows the temperature distribution in five slices of the substrate.

- 2 Open the **Plot Parameters** dialog box. Clear the **Slice** check box and select the **Boundary** check box. On the **Boundary** page select **hot** in the **Colormap** list. Click **OK** to see a boundary plot of the temperature.
- **3** Because the substrate is very thin you can get a better view of it by rescaling the *z*-axis. Select **Options>Axes/Grid Settings** and clear the **Axis equal** check box. After clicking **OK** to close the dialog box, click the **Zoom Extents** button on the Main toolbar.
- **4** To see what goes on inside the silicon substrate, hide the boundaries that are in the way. To do so, open the **Suppress Boundaries** dialog box from the **Options** menu and suppress Boundaries 1, 2, 4, 5, and 12. Then click the **Postprocessing Mode** button once again.
- 5 To see the temperature distribution along the trajectory, open the Domain Plot Parameters dialog box from the Plot menu and pick a time from the Solutions to use list on the General page. On the Line/Extrusion page select Edges 13, 16, and 19. Click Apply to see the plot.

Heat-Sink Experiments Using the Component Library

Introduction

The most common method for investigating a process or system is to first study the individual components in detail. You would probably start by setting up detailed models for the various components and then try to merge them in some way to describe the system as a whole. With COMSOL Multiphysics you can merge individual models to investigate how they interact as components in a system.

This exercise illustrates the use of the component library to merge different types of heat sinks with a heating device. The resulting models give an estimate of the thermal resistance of each heat sink and of the temperature distribution in the heating device. The model set up is similar to the benchmarking method for heat sinks used by Alpha Company Ltd. in Japan.

The models defined here are rather simple as they use a semi-empirical expression for the heat transfer coefficient between the heat sink and the air that flows over it. It is also possible to simulate the air flow in detail and thus the coupled heat transfer-flow problem in COMSOL Multiphysics, but this is outside the scope of this specific example.

The purpose of this example is to guide you through the procedure for merging different components into a single model of a system.

Model Definition

In this study, four different heat sinks from Alpha Company Ltd. are combined with one heating device. All the heat sinks have a base of 19 mm-by-19 mm and a height of



10 mm. The following figure shows the geometry of the heat sinks, all of which are made of aluminum 6063.

Figure 8-7: The four heat sinks are the types lpd19, n19, s1519, and ub19, all from Alpha Company Ltd.

The heating device measures 12.7 mm-by-12.7 mm and is 3 mm thick. It consists of an aluminum block with three heating elements made of steel. The next figure shows its geometry.



Figure 8-8: The heating device consists of three heating elements embedded in an aluminum block.

The physics inside the heat sinks is described by a heat balance where heat flux takes place by conduction. The thermal conductivity of aluminum 6063 is available in the material properties database in COMSOL Multiphysics.

The boundary condition at the top surface of the heat sinks sets the heat flux perpendicular to the boundary proportional to the temperature difference between the heat sink and the surrounding air according to the following equation:

$$-k\nabla T \cdot \mathbf{n} = h(T - T_{amb}). \tag{8-1}$$

The heat transfer coefficient, h, is obtained using a semi-empirical expression as a function of temperature and distance from the leading edge of the heat sink, in this case as a function of x in the next figure. In the previous equation, k denotes the thermal conductivity, T equals the temperature, and \mathbf{n} is the normal vector to the boundary. The boundary conditions at the bottom of the heat sink describe an influx of heat from the device being cooled.

The physics in the heating device is described by a heat balance, with the flux given by conduction, and with a heat-source term in each of the heating elements. The heating elements are made of steel, while the aluminum block is made of aluminum 6063. The thermal transport properties are available in the material properties database.

The boundary conditions describe thermal insulation at all boundaries except for the top surface, which is in contact with a heat sink. At this boundary the temperature is equal to that of the heat sink.

The top surface of the heating device has the dimensions of 12.7 mm-by-12.7 mm, while the bottom surface of the heat sink is 19 mm-by-19 mm. After the merger of one of the heat sinks and the heating device, the part of the bottom surface of the heat sink

that does not make contact with the heating device is thermally insulated as shown in the next figure.



Figure 8-9: The surface is the part of the heat sink that is insulated.

Once the two components are merged, all physics and boundary conditions are obtained from the components except for the insulation boundary condition shown in Figure 8-9. This boundary condition is actively defined in the model of the entire system.

Results

Figure 8-10 shows the calculated thermal resistance for the four heat sinks for the models when they are merged to the heating device. The thermal resistance is calculated as

Thermal resistance =
$$\frac{(T_{\text{contact}} - T_{\text{amb}})}{\text{Power}}$$
 (8-2)

where T_{contact} denotes the temperature in the middle of the contact surface between the heat sink and the heating device, and Power equals the total heating power of 5 W. The definition of the thermal resistance is taken from Alpha Company, Ltd. The models seem to overestimate the thermal resistances of the n19 heat sink but are otherwise in decent agreement with the experiments reported by Alpha Company, Ltd.



Figure 8-10: Thermal resistance as a function of the rate of air flowing over the heat sink.

The next figure shows the temperature distribution in the lpd19 heat sink in the cases where it is modeled by itself with a constant heat flux at the bottom surface, and then when the heating device is attached at the bottom. In both cases the heating power is 5 W and the flow rate of the cooling air is 4 m/s. The results are very similar except for the almost circular-shaped isosurface that is present in the merged model. In all cases the thermal resistance is slightly larger for the heat sink when attached to the heating device compared to the standalone model, where a constant heat flux is set at the base surface. This is an expected behavior because the flux is limited to the contact surface in the model of the entire system.



Figure 8-11: Temperature distribution in the standalone model of a lpd19 heat sink and in the model merged with the heating device (right).

The n19 heat sink displays a higher thermal resistance, which results in a maximum temperature of roughly 329 K, which is slightly higher than that for the lpd19 heat sink. The s1519 heat sink displays a lower resistance than that of the n19 but a resistance higher than that for the lpd19. The heat sink denoted ub19 is the one with

the best cooling performance in this study. Under the same conditions as shown in the previous figure it gives a maximum temperature of approximately 317 K.

In experiments performed by Alpha Company Ltd. the difference in performance between the lpd19, n19, and s1519 heat sinks is very small, smaller than the simulations in this study. However, this might be caused by the fact that the n19 and s1519 heat sinks are better inducers of turbulence, which improves the cooling performance. This is in line with the measured pressure losses, which are higher for the n19 and s1519. In addition, both of these models show a nonlinear dependence on the pressure drop as a function of flow, a behavior typical for turbulent flow. The experiments and the models both agree that the ub19 heat sink has the best performance under the testing conditions.

Model Library path: Component_Library/lpd19_heat_sink

Modeling Using the Graphical User Interface

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator select 3D in the Space dimension list.
- 3 Choose the application mode Comsol Multiphysics>Heat Transfer>Conduction.
- 4 Click OK.

GEOMETRY MODELING

- I From the File menu select Import>CAD Data From File.
- 2 Choose the COMSOL Multiphysics file file type and select the file lpd19_geo.mphbin. Click Import.

OPTIONS AND SETTINGS

I Open the **Options>Constants** dialog box. Enter the following constants for later use (the descriptions are optional); when done, click **OK**:

NAME	EXPRESSION	DESCRIPTION
Tam	25[degC]	Ambient temperature
mass_norm	3.590681[cm^3]*2700[kg/m^3]	Mass of clad
Uav	1[m/s]	Average velocity
dw	2.9[g]	Mass of device

NAME	EXPRESSION	DESCRIPTION
fact	mass_norm/(mass_norm-dw)	Velocity scaling
pow	5[W]	Power

2 Open the **Options>Expressions>Global Expressions** dialog box. Enter the following expressions (the descriptions are optional). Click **OK** when done.

NAME	EXPRESSION	DESCRIPTION
Lin	x+9.5005e-3	Inlet distance
Uavn	Uav*fact	Average scaled velocity
hcoef	10^(0.8616*log10((T+Tam)/2)-3.7142)/ Lin*(0.332*(0.1949*(log10((T+Tam)/ 2))^2-1.3494*log10((T+Tam)/ 2)+2.8469)^0.33*((350.9/T)*Uavn*Lin/ (-8E-12*T^2+4E-08*T+5E-06))^0.5)*flc2hs(5e5- (350.9/T)*Uavn*Lin/ (-8E-12*T^2+4E-08*T+5E-06),1e5)+(10^(0.8616* log10((T+Tam)/2)-3.7142))/ Lin*0.0296*((350.9/T)*Uavn*Lin/ (-8E-12*T^2+4E-08*T+5E-06))^0.8*(0.1949*(log 10((T+Tam)/2))^2-1.3494*log10((T+Tam)/ 2)+2.8469)^0.33*flc2hs((350.9/T)*Uavn*Lin/ (-8E-12*T^2+4E-08*T+5E-06)-5e5,1e5)	Heat transfer coefficient
powdens	pow/area	Heat flux

3 Open the Options>Integration Coupling Variables>Boundary Variables dialog box. Select Boundary 3 and create a boundary integration variable with Name area, Expression 1, Integration order 4, and Global destination. Click OK.

PHYSICS SETTINGS

Subdomain Settings

- I Open the **Subdomain Settings** dialog box from the **Physics** menu and select Subdomain 1.
- 2 Click the Load button to open the Materials/Coefficients Library dialog box.
- **3** Select **Aluminum 6063-T83** from the list and click **OK**.
- **4** Click the **Init** tab. In the **Initial value** area, enter Tam in the $T(t_0)$ edit field.
- 5 Click the Element tab. Then select Lagrange Linear from the Predefined elements list to use linear elements for the temperature field.
- 6 Click OK.

Boundary Conditions

- I Open the Boundary Settings dialog box from the Physics menu.
- 2 Enter the boundary conditions from the following table; when done, click **OK**.

SETTINGS	BOUNDARY 3	BOUNDARIES 1, 2, 4-118
Boundary condition	Heat flux	Heat flux
90	powdens	0
h	0	hcoef
T _{inf}	0	Tam

Notice that COMSOL Multiphysics indicates that the units for the hcoef and powdens expression variables are inconsistent if the highlighting of inconsistent units is active. This is due to the use of smoothing functions in the heat transfer coefficient (which require a nondimensional input) and the use of an integration coupling variable to compute the area. You can disregard these warnings.

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box from the Solve menu.
- 2 From the Solver list select Parametric.
- 3 Type Uav in the Parameter name edit field.
- **4** Type **1:0.2:4** in the **Parameter values** edit field.
- 5 Click **OK** to close the dialog box. Click the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To generate Figure 8-11 follow these instructions:

- I From the **Postprocessing** menu open the **Plot Parameters** dialog box. On the **General** page clear the **Slice** check box. Select the **Isosurface** check box.
- **2** Click the **Isosurface** tab. In the **Isosurface** levels area set the **Number of levels** to 10.
- **3** Click **OK** to close the dialog box and plot the isosurfaces.

Continue to model the second component, which is the heating device being cooled by the heat sink.

Model Library path: Component_Library/heating_device

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator select 3D in the Space dimension list.
- **3** In the list of application modes select

COMSOL Multiphysics>Heat Transfer>Conduction.

4 Click OK.

GEOMETRY MODELING

- I Select Work-Plane Settings from the Draw menu. Click OK.
- 2 Shift-click the **Rectangle/Square (Centered)** button on the Draw toolbar. In he **Width** field enter 12.7e-3, and set the **Base** to **Center**.
- 3 Click OK.
- 4 Click the **Zoom Extents** button on the Main toolbar to center the geometry.
- 5 From the Draw menu select Extrude. Set the Distance to -2e-3 and click OK.
- 6 Select Work-Plane Settings from the Draw menu. In the Plane area select the y-z option button. Click OK.
- 7 Click the Projection of All 3D Geometries button on the Draw toolbar.
- 8 Draw a circle of radius $0.7 \cdot 10^{-3}$ centered at the point (0, -0.001).
- 9 Select the circle by left-clicking on it. Press Ctrl+C to make a copy of the object.
- **IO** Press Crtl+V, set -3.5e-3 as its displacement in the x direction, and click **OK**.
- II Press Crtl+V again, and this time use 3.5e-3 as the x displacement. Click **OK**.
- 12 Select the three circles by holding down the shift key and clicking on each of them.
- 13 From the Draw menu select Extrude. Set the Distance to 6e-3 and click OK.
- I4 Click the Move button on the Draw toolbar. Set the distance in *x* direction to -3e-3, then click OK.
- IS Select all objects (press Ctrl+A) and then click the Union button on the Draw toolbar.

OPTIONS AND SETTINGS

I Open the **Options>Constants** dialog box. Enter the following constants for later use (the descriptions are optional); when done, click **OK**:

NAME	EXPRESSION	DESCRIPTION
htc	200[W/(m^2*K)]	Heat transfer coefficient
Tout	40[degC]	Outer temperature
power	5[W]/3	Power per heater

2 Open the Options>Integration Coupling Variables>Subdomain Variables dialog box.
 Select Subdomain 4 and create a subdomain integration variable with Name volume,
 Expression 1, Integration order 4, and Global destination. Click OK.

PHYSICS SETTINGS

Subdomain Settings

- I Open the **Subdomain Settings** dialog box from the **Physics** menu. Select Subdomains 2, 3, and 4.
- 2 Click the Load button to open the Materials Library dialog box.
- 3 Select Steel AISI 4340 from the list and click OK.
- 4 In the Heat source field enter power/volume.
- 5 Choose Subdomain 1 and click the Load button.
- 6 Select Aluminum 6063-T83 from the list and click OK.
- 7 Click the lnit tab. Select all subdomains and then, in the lnitial value area, enter Tout in the T(t₀) edit field to specify the initial value.
- 8 Click the Element tab and select Lagrange Linear for all subdomains.
- 9 Click **OK** to close the dialog box.

Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu. Enter the boundary conditions as in the following table; when done, click **OK**:

SETTINGS	BOUNDARY 4	BOUNDARIES I-3, 5
Boundary condition	Heat flux	Thermal insulation
9 ₀	0	
h	htc	
T _{inf}	Tout	

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

Now merge the two components into a single COMSOL Multiphysics model.

Model Library path: COMSOL_Multiphysics/Heat_Transfer/lpd19_dev_merge

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator select 3D in the Space dimension list.
- 3 Click OK.
- **4** From the **File** menu select **Open Component library**. Choose the **heating device** component and press the **Merge** button.
- **5** Press the **Merge** button a second time in the **Merge Component** dialog box. The component is now inserted into Geometry 1.
- 6 From the File menu select Open Component library. This time choose the **ldp19 heat sink** component and press the Merge button.
- 7 Press the **Merge** button again to merge the components into a a single geometry with a single application mode.
- 8 Mate the two objects by choosing Create pairs from the Draw menu, then click OK.

PHYSICS SETTINGS

Subdomain Settings

- I Open the Subdomain Settings dialog box from the Physics menu.
- 2 Click the **Init** tab. Select all subdomains and then, in the **Initial value** area, enter Tam in the **T(t₀)** edit field to specify the initial value
- 3 Click the Element tab and select Lagrange Linear for all subdomains.
- 4 Click **OK** to close the dialog box.

Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu. Select Boundary **3** and change the condition to **Thermal insulation**. Press **OK** to close the dialog box.

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box from the Solve menu.
- 2 From the Solver list select Parametric.
- 3 Type Uav in the Parameter name edit field.
- **4** Type **1:0.2:4** in the **Parameter values** edit field.
- 5 Click **OK** to close the dialog box, then click the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

To generate Figure 8-10 follow these steps:

- I From the Postprocessing menu open the Domain Plot Parameters dialog box. On the Point page select 226, then type (T-Tam) / pow in the Expression field.
- 2 Click the Line Settings button to choose Circle as Line marker, then select the Legend check box.
- 3 Click **OK** twice to close the dialog box and create the plot.
- **4** Repeat the merging process to create similar plots for the other three heat sinks from Alpha Company Ltd.: types n19, s1519, and ub19.

Multidisciplinary Models

9

Multidisciplinary models combine finite element analysis with other computational engineering tools and applications such as control system design, signal processing, state-space modeling, and dynamic simulations. Typically, the finite element model is embedded in the multidisciplinary model to detail physical processes.

Magnetic Brake

Introduction

A magnet brake in its simplest form consists of a disk of conductive material and a permanent magnet. The magnet generates a constant magnetic field, in which the disk is rotating. When a conductor moves in a magnetic field it induces currents, and the Lorentz forces from the currents slow the disk.



Model Definition

For a disk rotating with angular velocity ω about the *z*-axis, the velocity **v** at a point (x, y) is given by

$$\mathbf{v} = \boldsymbol{\omega}(-\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{0}).$$

When the disk is inserted in the air gap and it encounters the magnetic field \mathbf{B}_0 , the configuration induces a current density **j** according to Lorentz' equation

$$\begin{cases} \mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0) = \frac{1}{\sigma d} \mathbf{j} \\ \nabla \times \mathbf{E} = 0 \\ \nabla \times \mathbf{B} = \mu \mathbf{j}. \end{cases}$$

These equations use the following variables and material properties:

- **B** represents the magnetic field
- **E** is the electric field
- μ is the permeability
- σ is the electric conductivity
- *d* is the plate's thickness

This example uses the Dirichlet boundary condition $\mathbf{B} = 0$.

In this model, the magnetic flux \mathbf{B} has only a vertical component, and the currents and electric field have no *z*-components. Solving for \mathbf{B} gives the following scalar partial differential equation:

$$-\operatorname{div}\left(\nabla B + \mu \sigma d \omega (B + B_0) \begin{pmatrix} y \\ -x \end{pmatrix}\right) = 0$$

where B is the z-component of **B** and B_0 equals the z-component of **B**₀.

Now consider how the system evolves over time. The induced torque slows the disc down, and you must set up an ordinary differential equation (ODE) to model the angular velocity ω .

To obtain the time derivative of the angular velocity ω , compute the torque arising from the induced currents. For a small surface element, the force equals

$$\mathbf{dF} = \mathbf{j} \times (\mathbf{B} + \mathbf{B}_0) \mathbf{d}x \mathbf{d}y$$

and integrating over the disk gives the total torque:

$$\mathbf{M} = \int_{\text{Disk}} \frac{1}{\mu} \mathbf{r} \times \{ (\nabla \times \mathbf{B}) \times (\mathbf{B} + \mathbf{B}_0) \} dx dy.$$

In this case, **M** has only a *z*-component with the value

$$M = \int_{\text{Disk}} \frac{1}{\mu} \left(y \frac{\partial B}{\partial x} - x \frac{\partial B}{\partial y} \right) \cdot (B + B_0) dx dy.$$

Thus the ODE for the angular velocity ω can be formulated as

$$J\frac{d\omega}{dt} = M$$

where the moment of inertia J for a disk with radius r equals

$$J = m\frac{r^2}{2} = \frac{\rho dr^4\pi}{2}.$$

Modeling in COMSOL Multiphysics

The models show different ways to solve this problem:

- I Start by determining the magnetic field for a given angular velocity.
- **2** Continue by exporting to Simulink getting a time-dependent solution. Simulink models the additional ODE for the angular velocity of the disk. For each time step in Simulink, COMSOL Multiphysics solves a stationary PDE for the magnetic field given the current angular velocity.
- 3 Finish by doing the same thing entirely within COMSOL Multiphysics using coupling variables and an additional ODE for the angular velocity. Now COMSOL Multiphysics solves for the angular velocity and the magnetic field simultaneously. The discretized problem is a differential-algebraic equation system (DAE).

Results

There is agreement between the results from the Simulink simulation and the results from COMSOL Multiphysics. In the plots for ω and $d\omega/dt$ against time, $d\omega/dt$ is proportional to the torque on the disk.



Model Library path: COMSOL_Multiphysics/Multidisciplinary/ magnet_brake_simulink

Modeling Using the Graphical User Interface—Fixed w

MODEL NAVIGATOR

- I Select 2D in the Space dimension list.
- **2** In the list of application modes, open the **COMSOL Multiphysics>PDE Modes** folder and then the **PDE**, **Coefficient Form** node.
- 3 Select Stationary analysis.
- 4 Type B in the Dependent variables edit field.
- 5 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- **2** Enter the following constants:

NAME	EXPRESSION
К	mu*5.99e7*0.02
W	2*pi*200
J	8960*0.02*0.1^4*pi/2
mu	4e-7*pi
B0_2	0.1

The variable w represents the fixed angular velocity ω . The variable K is the product of μ , σ , and d.

GEOMETRY MODELING

- I In the Draw menu, point to Specify Objects and then click Circle.
- 2 In the **Circle** dialog box, type 0.1 in the **Radius** edit field and keep the center in the origin.
- 3 Click OK.
- 4 In the Draw menu, point to Specify Objects and then click Square.

- 5 In the Square dialog box, type 0.04 in the Width edit field. In the Position area, select Center in the Base list, and enter 0.05 and 0 in the x and y edit fields, respectively.
- 6 Click OK.
- 7 In the Draw menu, point to Specify Objects and click Point.
- 8 In the Point dialog box, type -0.1 and -0.1 in the x and y edit fields, respectively.
- 9 Click OK.

IO Click the **Zoom Extents** button.

You will use the point at (-0.1, -0.1) to introduce an additional scalar state variable corresponding to the angular velocity in the final version of this model.



PHYSICS SETTINGS

Boundary Conditions

Use the default Dirichlet boundary conditions.

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

2 Enter PDE settings as shown in the following table:

SETTINGS	SUBDOMAINS I, 2
с	1
a, f, d _a	0
α	K*w*y -K*w*x
γ	-K*w*B0*y K*w*B0*x

Expression Variables

Define a variable B_0 which is different in the two subdomains using an expression variable:

- I On the **Options** menu, point to **Expressions** and then click **Subdomain Expressions**.
- 2 Select Subdomain 1 and enter the name B0 in the Name column and 0 in the Expression column.
- **3** Select Subdomain 2 and enter the expression B0_2.
- 4 Click OK.

Coupling Variables

Define a coupling variable for the torque divided by the moment of inertia:

- I On the **Options** menu, point to **Integration Coupling Variables** and then click **Subdomain Variables**.
- 2 Select Subdomain 1 and 2 and enter the variable MdivJ in the Name column.
- **3** Enter the expression (B+B0)*(y*Bx-x*By)/mu/J. Keep the default integration order 4 and the default global destination.

	Name	Expression	Integration order	Global destination	
	MdivJ	(B+B0)*(y*Bx-x*	By) 4		1.
	-				
	-				
-	-				
act by group				V	
Select by group					1.,

4 Click OK.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box by selecting Free Mesh Parameters from the Mesh menu, click the Custom mesh size button and type in the 0.04 in the Maximum element size edit field.
- 2 Click the Subdomain tab.
- **3** Select Subdomain 2 and type 0.01 in Maximum element size.

Free Mesh Parameters	8
Global Subdomain Boundary Point Advanced	ОК
Subdomain selection Subdomain mesh parameters Maximum element size: 0.01 Element growth rate: Method: Triangle	Cancel Apply Help
Select Remaining Select Meshed	
Reset to Defaults Remesh Mesh Selected	

4 Click **OK** and then click the **Initialize Mesh** button.

COMPUTING THE SOLUTION

Click the **Solve** button to solve the problem.



I Clear the FEM structure, set the variable name and choose quadratic elements:

```
clear fem
fem.dim = 'B';
fem.shape = 2;
```

2 Specify the constants K, w, J, mu, and B0_2:

```
fem.const = {...
    'K','mu*5.99e7*0.02',...
    'w','2*pi*100',...
    'J','8960*0.02*0.1^4*pi/2',...
    'mu','4e-7*pi',...
    'B0 2',0.1};
```

3 Create the geometry:

```
fem.geom = circ2(0.1)+square2(0.04, 'pos', [0.05,0], 'base', 'cent');
```

4 Specify the boundary conditions: homogeneous Dirichlet conditions at the edge of the disk (boundary elements 5, 6, 7, and 8):

fem.bnd.h = 1; fem.bnd.r = 0; fem.bnd.ind = {5:8};

5 Specify the PDE coefficients:

```
fem.equ.c = 1;
fem.equ.al = {{{'K*w*y' '-K*w*x'}};
fem.equ.ga = {{{'-K*w*B0*y' 'K*w*B0*x'}};
fem.equ.expr = {'B0' {'0' 'B0 2'}};
```

6 Specify the integration coupling variable MdivJ from Subdomains 1 and 2:

```
clear el
el.elem = 'elcplscalar';
el.var = {'MdivJ'};
el.g = {'1'};
clear src
src.expr{1} = {'(B+B0)*(y*Bx-x*By)/mu/J'};
src.ipoints{1} = {'4'};
src.ind{1} = {'1' '2'};
el.src{1} = {{},{},src};
el.geomdim = {{}};
el.global = {'1'};
fem.elemcpl = {el};
```

7 Generate the mesh:

```
fem.mesh = meshinit(fem, 'hmax', 0.04, 'hmaxsub', [2 0.01]);
```

8 Solve the problem using the stationary solver and plot the solution:

```
fem.xmesh = meshextend(fem);
fem.sol = femstatic(fem);
postplot(fem,'tridata','B','tribar','on',...
  'trirefine',10,'axisequal','on');
```

Exporting to Simulink—Time-Dependent ω

This example begins in the same way as the previous model, except the solution of the time-dependent problem is obtained using Simulink instead of extended multiphysics.

Note: This section requires that you run COMSOL Multiphysics with MATLAB and Simulink.

Set up Problem for Fixed ω

- I Make sure you are running COMSOL Multiphysics with MATLAB, and that you have installed Simulink along with MATLAB.
- 2 First set up the problem for fixed ω as described from page 305 or on page page 309 if you are using the programming language.

Export the Model to Simulink

- I On the File menu, point to Export and then click Simulink Model to open the Export Simulink Model dialog box.
- 2 Select Export>Simulink on the File menu.
- **3** Enter magnet_struct as the model name. Use **General static** as the Simulink block type.
- 4 Click the Variables tab and type w B0_2 in the Variables edit field in the Input area. To add the variable dwdt in the origin, click Add, type dwdt in the Variable name edit
field, and make sure that the **Subdomain** button is selected. Type 0 and 0 in the **x** and **y** edit field, respectively, and type MdivJ in the **Expression** field.

This creates a structure that the COMSOL Multiphysics block in Simulink can use to run the COMSOL Multiphysics model as part of the Simulink simulation.

Export Simulink Model	X
General Variables	
Input Variables: w B0_2	
Output Variable number: 1 - Add Variable name: dwdt Variable definition: (a) Subdomain	Delete
x: 0 Predefined qua	ntities:
y: 0 Expression: Unit:	MdivJ
Point	
1 Predefined qua 2 E 3 Expression: 4 Unit:	ntties: B B V
	OK Cancel Apply

5 Open a Simulink model that can use the COMSOL Multiphysics model by typing magnet_break_mdl. As an alternative you can enter the Simulink model by following the instructions in the following section.

USING THE PROGRAMMING LANGUAGE

You can also perform the Simulink export on the MATLAB command line.

- Start by exporting the FEM structure to MATLAB by selecting Export>FEM
 Structure on the File menu. If you are using the programming language to enter the model you can ignore this step.
- **2** Then type the command.

```
magnet_struct = femsim(fem,'input',{'w' 'B0_2'},...
'output',{{'MdivJ' [0 0]'}},...
```

```
'outnames',{'dwdt'},'static','on')
```

This creates a structure which the COMSOL Multiphysics block within Simulink can use to run the COMSOL Multiphysics model as part of the Simulink simulation.

The syntax {'MdivJ' $[0 \ 0]$ '} means the output dwdt is computed by evaluating the expression MdivJ in the point (0, 0). Use the syntax of the postinterp command to specify outputs in this way.

3 Open a Simulink model that can use the COMSOL Multiphysics model by typing magnet_break_mdl. As an alternative you can enter the Simulink model by following the instructions in the following section.

DESIGN IN SIMULINK

Start Simulink and draw a model as in the following figure. You find the **COMSOL Multiphysics Subsystem** block in the **COMSOL Multiphysics** block library. To set up the input and the output ports, double-click the **COMSOL Multiphysics Subsystem** block and enter the **Simulink Structure** name magnet_struct.



- I Set the initial condition for the Integrator block to 2*pi*200.
- 2 In the Simulation Parameters dialog box set the solver type to ode15s, set the relative tolerance to 1e-2 and the absolute tolerance to 1e-3.
- **3** Start the simulation.

The scopes show graphs of ω and $d\omega/dt$ against time that correspond to the plots on page 304.

Model Library path: COMSOL_Multiphysics/Multidisciplinary/ magnet_brake

Modeling Using the Graphical User Interface —Time-Dependent ω

The modeling approach in this section uses the extended multiphysics feature to solve the time-dependent problem. Introduce a new variable w, active at a single point within the geometry using a Weak Form, Point application mode.

USING THE PREVIOUS MODEL

First set up the problem for fixed ω as described from page 305, but click the **Refine Mesh** button on the Main toolbar for a finer mesh. Without the export to Simulink, you can afford a finer mesh that increases the accuracy.

MODEL NAVIGATOR

- I Open the Model Navigator from the Multiphysics menu.
- 2 In the list of application modes, open the COMSOL Multiphysics>PDE Modes folder, and select Weak Form, Point.
- **3** Type w in the **Dependent variables** edit field.
- 4 Click Add.
- 5 Click OK.

OPTIONS AND SETTINGS

- I Open the **Constants** dialog box from the **Options** menu.
- I Select the variable w in the list and rename it w0. Later on, use this constant as initial value for w, the angular velocity.
- 2 Click OK.

PHYSICS SETTINGS

Point Settings

- I Open the Point Settings dialog box from the Physics menu.
- 2 Click the Init tab.
- **3** Select Point 1 and enter w0 as initial value in the w(t₀) edit field.
- 4 Click the Weak tab.

5 Enter PDE settings as shown for the weak point mode.

WEAK TERM	POINT I
weak	w_test*MdivJ
dweak	w_test*w_time
constr	0

6 Select Points 2 through 9 and clear the **Active in this domain** check box.

7 Click OK.

Defining Coupling Variables

The coupling variable MdivJ that integrates the torque has already been set up. You need to add a coupling variable that makes the angular velocity, w, available throughout the domain.

- I On the Options menu, point to Extrusion Coupling Variables and then click Point Variables.
- 2 Select Point 1 and enter the variable name w and the expression w.

Point selection	Name	Expression	
2 3 4 5 6 7 8	w	W	
9	O Linear transform General transform	nation	

- **3** Click the **General transformation** button.
- **4** Click the **Destination** tab.

5 Select **Subdomain** in the **Level** list and then select the check boxes for Subdomains 1 and 2.

oint Extrusio	on Variab	les							2
Source De	stination	Source	Vertices	Destination Vertices					
Geometry:	*Geom1	-	Variable:	w -					
evel:	*Subdom	nain 👻							
Subdomair	n selection	•	Vse s	elected subdomains as	destination				
Select	by group	0						1941	
						ОК	Cancel	Apply	Help

6 Click OK.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box and select Time dependent in the Solver list.
- 2 Type 0:0.2:8 in the Times edit field and then type 1e-3 in the Relative tolerance edit field and 1e-5 in the Absolute tolerance edit field.
- **3** Click the **Time Stepping** tab.
- **4** Select the **Time steps from solver** in the **Times to store in output** list.
- 5 Click OK.
- 6 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

Using the solution data, you can visualize the results in a number of ways:

- Click the Animate button to create an animation of the dissipation of *B* over time.
- For a graph of the ω against time, open the **Cross-Section Plot Parameters** dialog box from the **Postprocessing** menu. On the **General** page, select all the time steps in the

list and select **Point plot**. Click the **Point** tab and type w in the **Expression** edit field. Click **OK**.

Domain Plot Parameters		8
General Surface Line/I	Extrusion Point	
 Point plot y-axis data Predefined quantities: Expression: Unit: 	N v N v	
Point selection 2 3 4 5 6 7	x-axis data Auto Expression	
Line Settings	OK Cancel Apply Help	>

 For a graph of dω/dt against time, follow the same procedure as for ω but type MdivJ in the Expression edit field instead.

Process Control Using a PID Controller

In the chemical process industry it is often important to control a specific process. PID control (proportional-integral-derivative-control) is one way to achieve that, but it can be difficult to optimize the parameters in the PID algorithm. This example illustrates how you can implement a PID control algorithm to simulate a process control system and to find the optimal PID parameters.

This model is a generic example but could resemble the environment in a combustion chamber where the concentration at the ignition point is crucial. Two gas streams with different oxygen concentrations are mixed in the combustion chamber. The concentration is measured at the ignition point before complete mixing of the streams is reached. The control algorithm alters the inlet velocity of the gas with the lower oxygen content to achieve the desired total concentration at the ignition point.

Model Definition

The model geometry appears in the following figure. At the upper inlet, a gas stream with high oxygen content enters the reactor at a velocity of 10 mm/s, while a gas with a lower oxygen level enters from the left. The oxygen concentration is measured at a

measurement point, and the inlet velocity of the less concentrated stream is altered by the PID control algorithm to achieve the desired concentration at that point.



The model uses the Navier-Stokes equations to describe the fluid flow and the Convection and Diffusion application mode for the mass balance.

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot [\eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^{T})] + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{F}$$
(9-1)
$$\nabla \cdot \mathbf{u} = 0$$
$$\delta_{ts} \frac{\partial c}{\partial t} + \nabla \cdot (-D\nabla c) = R - \mathbf{u} \cdot \nabla c$$
(9-2)

It is possible to formulate the boundary conditions for the mass-transport equation by assuming that you know the two inlet concentrations. In addition, assume that the reactant transport at the outlet is mainly driven by convection, that is, neglect diffusion in the main direction of the convective flow. An insulation/symmetry boundary condition describes all walls. The applied boundary conditions for the mass balance are:

BOUNDARY	CONSTRAINT
Upper inlet	$c = c_{\rm in,top}$
Controlled inlet	$c = c_{\text{in,inlet}}$
Outlet	$\mathbf{n} \cdot (-D\nabla c) = 0$
Walls	$\mathbf{N}\cdot\mathbf{n}=0$

where *c* is the concentration; $c_{in,top}$ and $c_{in,inlet}$ are the inlet concentrations (mol/m³) for the upper and controlled inlets, respectively; *D* is the applied diffusivity (m²/s); and **N** is the molar flux (mol/(m²·s)).

The model uses the following boundary conditions for the fluid flow:

BOUNDARY	CONSTRAINT
Upper inlet	$\mathbf{u} = (0, -v_{\text{in,top}})$
Controlled inlet	u = $(u_{in}, 0)$
Outlet	$p_0 = 0$
Inlet sections	$\mathbf{n} \cdot \mathbf{u} = 0$
Walls	$\mathbf{u} = 0$

where **u** is the velocity vector (m/s), $v_{in,top}$ is the inlet velocity at the top inlet, and u_{in} is the PID controlled velocity. At the outlet, set the pressure to p_0 (Pa). No-slip boundary conditions describe all walls except the inlet sections where slip conditions apply, allowing for a smooth transition to a laminar velocity profile.

The PID control algorithm used to calculate u_{in} is defined as follows:

$$u_{\rm in} = k_{\rm P}(c - c_{\rm set}) + k_{\rm I} \int_{0}^{t} (c - c_{\rm set}) dt + k_{\rm D} \frac{\partial}{\partial t} (c - c_{\rm set})$$
(9-3)

using the following parameters:

PARAMETER	VALUE
$c_{ m set}$	0.5 mol/m ³
k_{P}	0.5 m ⁴ /(mol·s)
k_{I}	I m ⁴ /(mol·s ²)
k_{D}	10 ⁻³ m ⁴ /mol

In practice, the derivative constant, $k_{\rm D}$, is set to 0 in most cases as this parameter can be difficult to determine and the term may increase the fluctuations in the system.

Results

The two plots in Figure 9-1 show the oxygen concentration and the velocity stream lines in the chamber after approximately 0.04 s and 1.4 s, respectively. The figures show that the measured concentration depends strongly on the flow field. At start-up,

when the inlet velocity of the stream entering from the left is very low, the sensor is entirely exposed to the highly concentrated stream, and as the inlet velocity increases the opposite relation occurs.



Figure 9-1: Oxygen concentration and velocity streamlines after approximately 0.04 s (top) and 1.4 s (bottom).

Figure 9-2 shows the inlet velocity and concentration in the measurement point as a function of time for two different values for the $k_{\rm P}$ parameter. The solid line represents the results for a $k_{\rm P}$ value of 0.5 m⁴/(mol·s) while the dashed line corresponds to $k_{\rm P}$ equal to 0.1 m⁴/(mol·s). The results evaluated for the smaller $k_{\rm P}$ value oscillate more



before stabilizing. Thus, it is clear that for this case the higher $k_{\rm P}$ value yields a more stable process control.

Figure 9-2: PID-controlled inlet velocity (tpp) and concentration in the measurement point as a function of time for $k_{\rm P} = 0.5~m^4/(mol\cdot s)$ (solid) and $k_{\rm P} = 0.1~m^4/(mol\cdot s)$ (dashed).

Model Library path: COMSOL_Multiphysics/Multidisciplinary/PID_control

MODEL NAVIGATOR

- I Go to the Model Navigator and select 2D in the Space dimension list.
- 2 In the list of application modes select COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis.
- 3 Click the Multiphysics button in the lower right corner, then click Add.
- 4 Similarly, add the COMSOL Multiphysics>Convection and Diffusion>Convection and Diffusion>Transient analysis application mode.
- 5 Click OK.

OPTIONS AND SETTINGS

Constants

- I Go to the Options menu and choose Constants.
- 2 Make the following entries in the Constants dialog box; when finished, click OK.

NAME	EXPRESSION	DESCRIPTION
v_in_top	0.01[m/s]	Velocity, upper inlet
cin_top	1[mol/m^3]	Concentration, upper inlet
cin_inlet	0.2[mol/m^3]	Concentration, controlled inlet
c00	0.5[mol/m^3]	Initial concentration, chamber interior
eta	3e-5[Pa*s]	Dynamic viscosity
rho	1.2[kg/m^3]	Density
D	1e-4[m^2/s]	Diffusivity
c_set	0.5[mol/m^3]	Setpoint concentration
k_P_ctrl	0.5[m^4/(mol*s)]	Proportional parameter
k_I_ctrl	1[m^4/(mol*s^2)]	Integral parameter
k_D_ctrl	1e-3[m^4/mol]	Derivative parameter

Grid Settings

I Go to the **Options** menu and select **Axes/Grid Settings**.

AXIS	VALUE
x min	-0.015
x max	0.015
y min	-0.01
y max	0.02

2 On the **Axis** page enter **x-y limits** (expressed in the default length unit meters) for the axes as displayed below.

3 Click the **Grid** tab.

4 Clear the Auto check box and enter the following extra grid values.

GRID	VALUE
Extra x	-0.014 -0.012 -0.01 -0.004 -0.002 0 0.002 0.008 0.01 0.012
Extra y	-0.004 -0.002 0 0.002 0.004 0.008

5 Click OK.

GEOMETRY MODELING

When creating the geometry you alternate between using the Line and 2nd Degree
 Bézier Curve tools located on the Draw toolbar to the left. For curved shapes use the curve tool, and for straight lines use the line tool. Click on the points given in the following table using the corresponding drawing tool. Use the coordinates displayed in the lower-left corner of the screen to find the points.

DRAW TOOL	POINTS (X, Y)
Line	(0.012,-0.002) (0.012,-0.004) (-0.01,-0.004)
Curve	(-0.01,-0.002) (-0.012,-0.002)
Line	(-0.014,-0.002) (-0.014,0) (-0.012,0)
Curve	(-0.01,0) (-0.01,0.002)
Line	(-0.004,0.002)
Curve	(-0.002,0.002) (-0.002,0.004)
Line	(-0.002,0.008) (0,0.008) (0,0.004)
Curve	(0,0.002) (0.002,0.002)
Line	(0.008,0.002) (0.008,0)
Curve	(0.008,-0.002) (0.01,-0.002)
Line	Right-click to close the geometry

- **2** To add the measurement point, click the **Point** button in the Draw toolbar to the left and click on the coordinates $(0, -2 \cdot 10^{-3})$.
- 3 Click the Zoom Extents button on the Main toolbar to center the geometry.

PHYSICS SETTINGS

Subdomain Settings—Incompressible Navier-Stokes

- I Go to the Multiphysics menu and select Incompressible Navier-Stokes (ns).
- 2 From the Physics menu, choose Subdomain Settings.
- **3** In the **Subdomain Settings** dialog box select Subdomain 1 and type rho in the **Density** edit field.
- 4 In the **Dynamic viscosity** edit field type eta.
- 5 Click OK.

Boundary Conditions-Incompressible Navier-Stokes

- I From the Physics menu, choose Boundary Settings.
- 2 In the Boundary Conditions dialog box enter the following boundary conditions.

SETTINGS	BOUNDARY I	BOUNDARY 7	BOUNDARIES 2, 3, 6, 8	BOUNDARIES 4, 5, 9–11, 13–17	BOUNDARY 12
Туре	Inlet	Inlet	Symmetry	Wall	Outlet
Condition	Velocity	Velocity	-	No slip	Pressure, no viscous stress
u ₀	u_in_ctrl* (u_in_ctrl>=0)	0			
v ₀	0	-v_in_top			
Po					0

You will define u_in_ctrl as a scalar expression shortly.

3 Click OK.

Subdomain Settings—Convection and Diffusion

- I From the Multiphysics menu, choose Convection and Diffusion (cd).
- 2 From the Physics menu, choose Subdomain Settings select Subdomain 1.
- **3** In the **Diffusion (isotropic)** edit field type D to define an isotropic diffusion coefficient.
- 4 In the **x-velocity** edit field type u.
- 5 In the y-velocity edit field type v.

- 6 Click the **Init** tab.
- 7 In the c(t₀) edit field enter the initial concentration c00.
- 8 Click OK.

Boundary Conditions—Convection and Diffusion

- I From the Physics menu, choose Boundary Settings.
- 2 In the Boundary Conditions dialog box enter the following boundary conditions.

SETTINGS	BOUNDARY I	BOUNDARIES 2-6, 8-11, 13-17	BOUNDARY 7	BOUNDARY 12
Туре	Concentration	Insulation/ Symmetry	Concentration	Convective flux
c ₀	cin_inlet		cin_top	

3 Click OK.

PID ALGORITHM SETUP

It is possible to specify the PID algorithm as a scalar expression. However, some of the terms in the algorithm require some additional specifications. The algorithm contains the concentration in the measurement point, the time derivative of the concentration in the point, and the time integration of the concentration difference between the measured and wanted concentration.

- I Go to the **Options** menu and choose **Expressions>Scalar Expressions**.
- 2 Set the Name to u_in_ctrl and define the Expression as

nojac(k_P_ctrl*(conc-c_set)+k_I_ctrl*Int+k_D_ctrl*ctime). The nojac operator makes sure the above expression gives no Jacobian contribution. The resulting effect is the same as using nonideal weak constraints, namely a one-way coupling instead of the normal two-way coupling. Because of the use of a coupling variable, the unit becomes undefined for this variable and other quantities where you use it.

- 3 Click OK.
- **4** To specify the concentration and its time derivative in the point, go to the **Options** menu and choose **Integration Coupling Variables>Point Variables**.
- **5** Select Point 10 and enter conc in the **Name** edit field and c in the **Expression** edit field.
- 6 For the same point enter ctime in the Name edit field and ct in the Expression edit field. ct is the predefined expression for the time derivative of the concentration.

- 7 Click OK.
- 8 To specify the time integral of the concentration difference in the point, go to the Physics menu and open the Space-Independent Equations dialog box.
- 9 Enter Int in the Name (u) edit field.
- **IO** In the **Equation** edit field type Intt-(conc-c_set).
- II Click OK to exit the Space-Independent Equations dialog box.

MESH GENERATION

- I Go to the Mesh menu and choose Free Mesh Parameters.
- 2 On the Global tab, select Finer in the Predefined mesh sizes list.
- **3** To reduce the number of elements along the curved boundaries, click the **Custom mesh size** button and type 0.5 in the **Mesh curvature factor** edit field.
- 4 Click **Remesh** and **OK**.

COMPUTING THE SOLUTION

- I Go to the Solve menu and select Solver Parameters.
- 2 Select the Transient from the Analysis list.
- **3** On the **General** page, set the **Times** vector to 0:0.04:6.
- **4** On the **Time Stepping** page, select **Time steps from solver** from the **Times to store in output** list to make all time steps the solver takes available for postprocessing.
- **5** Select **Intermediate** from the **Time steps taken by solver** list to give the solver some freedom to set its own time steps. Using this option restricts the solver to take at least one time step in each subinterval of the time list.
- 6 Click OK to exit the Solver Parameters dialog box.
- 7 Click the Solve button on the Main toolbar to start the simulation.

POSTPROCESSING AND VISUALIZATION

The default visualization plots the magnitude of the velocity field at the end of the simulation interval. Follow the instructions below to reproduce the simultaneous plots of the oxygen concentration and the velocity stream lines at two different times presented in Figure 9-1 on page 320.

- I Go to the **Postprocessing** menu and choose **Plot Parameters** to open the **Plot Parameters** dialog box.
- **2** On the **General** page, select the solution near t = 0.04 s from the **Solution at time** list.
- 3 On the Surface page, click the Surface Data tab.

- 4 In the Predefined quantities list, select Convection and Diffusion (cd)>Concentration, c.
- **5** Click the **Streamline** tab.
- 6 Select the Streamline plot check box. In the Predefined quantities list on the Streamline Data tab keep the default selection, Incompressible Navier-Stokes (ns)>Velocity field.
- 7 From the Streamline plot type list select Magnitude controlled.
- 8 Click **Apply** to view the upper plot in Figure 9-1.
- 9 Return to the General tab, then select the solution at t = 1.4 s from the Solution at time list. Click OK to generate the lower plot in Figure 9-1.

To visualize the inlet velocity and the measured concentration as functions of time, use domain plots:

- I Open the Domain Plot Parameters dialog box from the Postprocessing menu.
- 2 Click the **Point** tab.
- 3 Select Point 1 and enter u_in_ctrl*(u_in_ctrl>=0) in the Expression edit field.
- 4 Click Apply.
- 5 Click the General tab and select to plot in New figure in the Plot in list.
- 6 Click the Point tab, select Point 10, and select Concentration, c from the Predefined quantities list.
- 7 Click OK.

To see the above plots for different values of $k_{\rm P}$, alter the $k_{\rm P}$ value and solve the model again without closing the plot frames:

- I Open the Constants dialog box from the Options menu and change the value of k_P_ctrl to 0.1 [m^4/(mol*s)].
- 2 Click the **Solve** button to solve the model for the new parameter value.
- **3** Open the **Domain Plot Parameters** dialog box from the **Postprocessing** menu.
- **4** On the **General** page, select to plot in **Figure 2** and select the **Keep current plot** check box.
- **5** Click the **Point** tab and make sure Point 10 and **Concentration**, **c** are selected.
- 6 Click the Line Settings button. Set the Line style to Dashed line and click OK.
- 7 Click Apply to generate the lower plot in Figure 9-2 on page 321.
- 8 Click the General tab, select to plot in Figure 1, and select the Keep current plot check box.

- 9 Click the Point tab, select Point 1, and type u_in_ctrl*(u_in_ctrl>=0) in the Expression edit field.
- **IO** Click **OK** to close the dialog box and generate the upper plot in Figure 9-2.

Controlling Temperature

Introduction

This example demonstrates two approaches to controlling the temperature in a heated metal block using a thermal controller. The first model uses an on-off controller. You make this model in COMSOL Multiphysics and export it to Simulink. The second approach uses a PI controller to control the temperature.

Model Definition

The study of this thermal controller involves two distinct modeling situations:

- Taken as a whole, this dynamic system assumes only a small number of states, making it a good candidate for modeling and simulation with Simulink.
- One of the model's elements, the controller, contains a subsystem that involves heat distribution, which is described with a PDE, the heat equation. It is easy to break this portion of the problem down and model it in COMSOL Multiphysics.

The dynamic system consists of a metal block that exchanges heat with the environment. A heater and a thermostat switch are situated inside the glass-enclosed system. The system works as follows: The thermostat turns the heater on or off when the temperature becomes too low or too high.



The finite-element model of the metal block requires two inputs:

- The state of the heater, which can be On (1) or Off (0)
- The exterior temperature, T_{out}

As its output, the model supplies the temperature at the thermostat's location.

The PDE describes the overall system's temperature distribution given the temperature of the heater and the exterior environment. If the heat transfer is so fast that the heat distribution is more or less constant (in space, not in time), a single state is sufficient. Otherwise, controlling the temperature requires modeling a PDE in COMSOL Multiphysics and then including those results in the Simulink environment as illustrated here.

The heat equation is:

$$\rho C \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q.$$

The boundary conditions come from the level of insulation around the system. On well-insulated sides the temperature flux is zero, which gives the Neumann boundary condition $\mathbf{n} \cdot (k\nabla T) = 0$. The poorly insulated sides involve the Neumann condition $\mathbf{n} \cdot (k\nabla T) = k_g/l_g \cdot (T_{\text{out}} - T)$, where k_g and l_g are the thermal conductivity and the thickness of the glass sheet that separates the metal block and the exterior.



Because only temperature distribution in the xy-plane are of interest, you can use a 2D model. For the units to make sense, think of the domain as having a depth (z direction) of 1 m.

Modeling in COMSOL Multiphysics

This model is based on the SI system, but to more appropriately represent the time scale of the problem, the time is counted in minutes. This means that you must scale all quantities with units that contain the basic unit of time by a factor 60 compared to

their standard SI values. This model data in the following lists includes this conversion for the thermal conductivity k and the heat source Q.

Metal block:

- Dimensions $30 \times 20 \times 100$ cm = $0.3 \times 0.2 \times 1$ m
- Density $\rho = 7.82 \cdot 10^3 \text{ kg/m}^3$
- Heat capacity C = 449 J/(kg·K)
- Thermal conductivity $k = 82 \text{ W}/(\text{m}\cdot\text{K}) = 4.92 \cdot 10^3 \text{ J}/(\text{m}\cdot\text{min}\cdot\text{K})$

Glass sheet:

- Thermal conductivity $k_g = 0.9 \text{ W/(m \cdot K)} = 54 \text{ J/(m \cdot min \cdot K)}$
- *Thickness* $l_g = 1 \text{ mm} = 10^{-3} \text{ m}$

Heater:

- Dimensions $4 \times 4 \times 100$ cm = $0.04 \times 0.04 \times 1$ m
- Heat source $Q = 2 \text{ kW/V}_{\text{heater}} = 2 \cdot 10^3 / (1 \cdot 0.04^2) \text{ W/m}^3 = 7.5 \cdot 10^7 \text{ J/(m}^3 \cdot \text{min)}$

Do the modeling in COMSOL Multiphysics in three steps:

- I Start by setting up a basic model of the steel plate for the Simulink export.
- 2 Continue by making the model of the on-off controller in Simulink.
- **3** Finish by using a PI controller in a model implemented entirely within COMSOL Multiphysics using coupling variables and the Weak Point application mode to add the ODE.

Results

The first model uses Simulink to simulate the on-off controller. Simulink handles the nonlinear behavior of the on-off controller well. The PDE model for the plate is linear, you can do the Simulink export using the linearized state-space form to improve performance. You can also use the mode reduction facility of the Simulink export to improve performance in Simulink, but it only works for linear or linearized PDE models.

It is easy to create the second model entirely within COMSOL Multiphysics, modeling the PI controller with an ODE.

The first set of images show the result of the Simulink simulation with the on-off controller. The left figure shows the thermostat temperature at node 3 and the right figure shows the result when the thermostat has been move to node 8.



Notice a few interesting details in the solutions. For instance, the distance between the thermostat and the exterior has an effect on the delay between the peaks of the two curves. Node 8 is further from the exterior than node 3, so the delay is larger in the second plot than in the first.

Notice also the qualitative differences between the two simulations. This extra information comes as a bonus for solving a detailed heat-flow problem that accounts for the geometry and the heat-diffusion properties.

The second set of images show the result of the simulation of the PI-controlled metal plate, illustrating the ability of the PI controller to keep the temperature close to 20 degrees. The left figure shows the thermostat temperature, the right figure shows the external temperature, and the last figure shows the controller output level.





Model Library path: COMSOL_Multiphysics/Multidisciplinary/ thermal_controller_simulink

Using the Graphical User Interface—No Controller

Before exporting the model description to Simulink, run a time-dependent simulation in COMSOL Multiphysics, running with the heater turned on with an initial temperature of 20 degrees in the plate.

MODEL NAVIGATOR

- I Select 2D in the Space dimension list.
- 2 In the list of application mode, open COMSOL Multiphysics>Heat Transfer and then select Conduction and finally Transient analysis.
- **3** Use the default dependent variable name T. Make sure Lagrange Quadratic elements are selected in the Elements list.
- 4 Click OK.

GEOMETRY MODELING

- I In the Draw menu, point to Specify Objects and then click Rectangle.
- 2 In the Rectangle dialog box, type 0.3 and 0.2 in the Width and Height edit fields.
- 3 Click OK.
- 4 Click the Zoom Extents button.

- 5 In the Draw menu, point to Specify Objects and then click Square.
- 6 In the Square dialog box, type 0.04 in the Width edit field. In then Position area, select Center in the Base list and enter 0.1 and 0.1 in the x and y edit fields.
- 7 In the Draw menu, point to Specify Objects and then click Point.
- 8 In the **Point** dialog box, type 0.05 0.2 and 0.1 0.1 in the x and y edit fields, respectively, to create two points at (0.05, 0.1) and (0.2, 0.1).
- 9 Click OK.



OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- **2** Enter the following constants:

NAME	EXPRESSION
HeatState	1
Tout	20

3 Click OK.

PHYSICS SETTINGS

Boundary Conditions

I From the Physics menu, choose Boundary Settings.

2 Enter the following boundary conditions.

SETTINGS	BOUNDARY I	BOUNDARIES 2, 3, 8
Туре	Heat flux	Thermal insulation
h	54/1e-3	
T _{inf}	Tout	

3 Click OK.

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

2 Enter the following material properties.

SETTINGS	SUBDOMAIN I	SUBDOMAIN 2
k	4.92e3	4.92e3
ρ	7.82e3	7.82e3
C _p	449	449
Q	0	7.5e7*HeatState

3 Click the **Init** tab.

4 Enter the following initial values.

SETTINGS	SUBDOMAINS I, 2
T(t ₀)	20

5 Click OK.

MESH GENERATION

To speed up the computation within Simulink, use a coarse mesh.

- I Open the Free Mesh Parameters dialog box, click the Custom mesh size button and enter 0.05 in the Maximum element size edit field.
- 2 Click **OK** to close the dialog box.
- **3** Initialize the mesh.

COMPUTING THE SOLUTION

Solve the heat equation using 21 equal steps for 20 minutes, starting at time = 0:

- I Open the Solver Parameters dialog box from the Solve menu.
- 2 Enter 0:1:20 in the Times edit field.

- 3 Click OK.
- 4 Click the Solve button to start the simulation.

 Decision (grad How Internal Action (grad How Internal (grad How Interna (grad How Internal (grad How Internal (grad How Int

A plot of the surface temperature at Time = 20 appears as in this figure:

Exporting to Simulink—Temperature Control

The result of exporting a model from COMSOL Multiphysics to Simulink is a structure variable in the MATLAB workspace. To use the model in Simulink, simply drag the **COMSOL Multiphysics Subsystem** block to your Simulink model as described in "Modeling in Simulink" on page 337.

The following variables affect the temperature of the metal block:

- Exterior temperature, Tout
- The state of the heater, HeatState, which can have the values 1 or 0, corresponding to on or off

The output from the COMSOL Multiphysics model is the temperature in the block where the thermostat is located:

• Thermostat temperature, Temp

You need this variable to control the heater.

EXPORTING TO SIMULINK

I Enter the model as described on page 333.

- **2** On the File menu, point to Export and then click Simulink Model to open the Export Simulink Model dialog box.
- 3 Enter blocksct in Structure name and change Simulink block type to Linearized dynamic. On the Variables tab, enter HeatState Tout in Input. To add the output variable Temp, click Add and enter Variable name Temp. Click Point, and select point 3 in the list. Keep the Expression T. Click OK.
- **4** Open a Simulink model that can use the COMSOL Multiphysics model by typing in MATLAB

```
thermal_controller_mdl
```

As an alternative you can enter the Simulink model by following the instructions in the section "Modeling in Simulink" below.

USING THE PROGRAMMING LANGUAGE

- I Enter the model as described on page 333.
- **2** On the **File** menu, point to **Export** and then click **FEM Structure** to export the FEM Structure to MATLAB.
- 3 Type the command

```
blocksct = femsim(fem,'input',{'HeatState' 'Tout'},...
'output',{{'T' zeros(0,1) 'dom' 3}},...
'outnames',{'Temp'},'state','on');
```

This creates a structure which the COMSOL Multiphysics block within Simulink can use to run the COMSOL Multiphysics model as part of the Simulink simulation.

The syntax {'T' zeros(0,1) 'dom' 3} means the output Temp is computed by evaluating the expression T in point 3. Use the syntax of the postinterp command to specify outputs in this way.

4 Open a Simulink model that can use the COMSOL Multiphysics model by typing

thermal_controller_mdl

As an alternative you can enter the Simulink model by following the instructions below.

MODELING IN SIMULINK

I Start Simulink by typing simulink in the MATLAB Command Window. Draw a model as shown below. You find the **COMSOL Multiphysics Subsystem** block under

COMSOL Multiphysics in the Simulink block library. See the Simulink documentation for more information about creating models.



2 To set up the input and output ports of the COMSOL Multiphysics Subsystem block, double-click that block and set the structure name to blocksct. Click OK.

Block Parameters: FEMLAB Subsystem	X			
FEMLAB Subsystem (mask)				
Models a FEMLAB subsystem.				
The FEM Simulink Structure can be created with FEMSIM or with the "Export to Simulink" option in the FEMLAB Graphical User Interface.				
Parameters				
FEM Simulink Structure				
blockset				
	-			
OK Cancel <u>H</u> elp <u>Apply</u>				

3 Specify the values of the constant blocks for **Thermostat Setting** and **Average Exterior Temperature** as in the previous figure.

4 In the Relay block, enter the value 1 as the Switch on point and -1 as the Switch off point.

Block Parameters: Relay 🛛 🛛 🔀
 Relay Output the specified 'on' or 'off' value by comparing the input to the specified thresholds. The on/off state of the relay is not affected by input between the upper and lower limits.
Parameters Switch on point
Switch off point:
) Output when on: 1
Output when off:
Show additional parameters Sample time (-1 for inherited): -1
OK Cancel Help Apply

5 In the Sine Wave block, set the amplitude to 10 and the frequency to 2*pi/120, which corresponds to a period of 120 minutes.

Block Parameters: Variation of Exterior Temp
Sine Wave
Output a sine wave where the sine type determines the computational technique used. The parameters in the two types are related through:
Samples per period = 2*pi / (Frequency * Sample time)
Number of offset samples = Phase * Samples per period / (2 * pi)
Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
Parameters
Sine type: Time based
Amplitude:
10
Bias:
0
Frequency (rad/sec):
2*pi/120
Phase (rad):
0
Sample time:
0
✓ Interpret vector parameters as 1-D
OK Cancel Help Apply

6 Open the Simulation Parameters dialog box by making a selection from the Simulation menu. Set the Stop time to 240 minutes and the Solver to ode15s. The stability of discrete heat equations is known to put severe constraints on the time

step of explicit methods. Thus you should select the ode15s solver (implicit) rather than the explicit integrators.

🚺 Sim	ulation Parame	eters: thermal_controller 🔳 🗖 🗙
Solver	Workspace I/O	Diagnostics Advanced
- Simul Start	ation time time: 0.0	Stop time: 240
Solve Type	r options : Variable-step _▼	ode15s (stiff/NDF)
Max	step size: auto	Relative tolerance: 1e-3
Min s	tep size: auto	Absolute tolerance: auto
Initial	step size: auto	Maximum order: 5
Outp	ut options	í
Refi	ne output	Refine factor: 1
		OK Cancel Help Apply

- 7 Now solve the system by selecting **Start** from the **Simulation** menu.
- 8 To view the temperature at node 3, double-click the **Scope** block.
- 9 Finally switch the thermostat from node 3 to node 8. On the File menu, point to Export and then click Simulink Model to open the Export Simulink Model dialog box.
- **IO** Select point 8 in the list. Click **OK**.
- II Restart the simulation in Simulink.

10

Multiphysics Models

Multiphysics models include the coupling of several dependent variables from different physics in the same model. Typical multiphysics applications include fluid-structure, thermal-structure, and thermal-electric couplings. In COMSOL Multiphysics, you can also build extended multiphysics models, which contain several kinds of physics and equations on more than one geometry. Extended multiphysics model can include any mix of 0D (ODEs or algebraic equations), 1D, 2D, and 3D geometries.

Magnetic Drug Targeting in Cancer Therapy

Introduction

Current research on methods to target chemotherapy drugs in the human body includes the investigation of so-called bio-compatible magnetic nanocarrier systems. For example, magnetic liquids such as ferrofluids can play an important role as drug carriers in the human body (Ref. 1). As such, they can be used for drug targeting in modern locoregional cancer treatment. A remaining challenge for this medical application is the choice of clinical setting. Important parameters are optimal adjustment of the external magnetic field and the choice of ferrofluid properties.

Avoiding damage on healthy human cells from chemotherapy drugs, imposes an upper limit in the treatment dose. This limit impedes the chances of successful treatment of the tumor cells. One objective of modern cancer research is therefore to concentrate chemotherapy drugs locally to tumor tissue and to weaken the global exposure to the organism.

This model of the ferrohydrodynamics of blood demonstrates a simple setup for investigating an external magnetic field and its interaction with blood flow containing a magnetic carrier substance. The model treats the liquid as a continuum, which is a good first step. You can extend this model by particle tracing, making it a multiscale model. The equations and theory are based on Maxwell's equations and the Navier-Stokes equations. You first solve Maxwell's equations in the full modeling domain formed by permanent-magnet, blood-vessel, tissue, and air subdomains. A magnetic volume force, couples the resulting magnetic field to a fluid-flow problem in the blood-vessel domain described by the Navier-Stokes equations.

Model Definition¹

The model geometry represents a blood vessel, a permanent magnet, surrounding tissue, and air in 2D. Blood feeds into the vessel from the left in Figure 10-1. The velocity and pressure fields are calculated in the blood stream. COMSOL Multiphysics computes the magnetic field (magnetic vector potential) generated by the permanent

^{1.} This model was provided by Dr. Daniel J. Strauss, The Institute for New Materials, Inc., www.inm-gmbh.de.

magnet. This magnetic field generates a magnetic volume force that affects the flow field in the blood vessel.



Figure 10-1: Geometric representation of the model.

MAGNETOSTATIC EQUATIONS

Because the magnetic part of this problem is static, Maxwell-Ampere's law for the magnetic field \mathbf{H} (A/m) and the current density \mathbf{J} (A/m²) applies:

$$\nabla \times \mathbf{H} = \mathbf{J} \tag{10-1}$$

Furthermore, Gauss' law for the magnetic flux density ${f B}$ (Vs/m²) states that

$$\nabla \cdot \mathbf{B} = 0. \tag{10-2}$$

The constitutive equations describing the relation between **B** and **H** in the different parts of the modeling domain read:

$$\mathbf{B} = \begin{cases} \mu_0 \mu_{r, mag} \mathbf{H} + \mathbf{B}_{rem} & \text{permanent magnet} \\ \mu_0(\mathbf{H} + \mathbf{M}_{ff}(\mathbf{H})) & \text{blood stream} \\ \mu_0 \mathbf{H} & \text{tissue and air} \end{cases}$$
(10-3)

Here μ_0 is the magnetic permeability of vacuum (Vs/(A·m)); μ_r is the relative magnetic permeability of the permanent magnet (dimensionless); \mathbf{B}_{rem} is the remanent magnetic flux (A/m); and \mathbf{M}_{ff} is the magnetization vector in the blood stream (A/m), which is a function of the magnetic field, **H**.

Defining a magnetic vector potential A such that

$$\mathbf{B} = \nabla \times \mathbf{A}, \qquad \nabla \cdot \mathbf{A} = 0, \qquad (10-4)$$

you finally, by substitution in Equation 10-1 through Equation 10-3, arrive at the following vector equation to solve:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} - \mathbf{M}\right) = \mathbf{J}$$

Simplifying to a 2D problem with no perpendicular currents, this equation reduces to

$$\nabla \times \left(\frac{1}{\mu_0} \nabla \times \mathbf{A} - \mathbf{M}\right) = \mathbf{0}.$$
 (10-5)

Note that this equation assumes that the magnetic vector potential has a nonzero component only perpendicularly to the plane, $\mathbf{A} = (0, 0, A_z)$.

An arc tangent expression with two material parameters α (A/m) and β (m/A) characterizes the induced magnetization $\mathbf{M}_{\text{ff}}(x, y) = (M_{\text{ff}x}, M_{\text{ff}y})$ of a ferrofluid (Ref. 2):

$$M_{x} = \alpha \operatorname{atan}\left(\frac{\beta}{\mu_{0}} \frac{\partial A_{z}}{\partial y}\right)$$
$$M_{y} = \alpha \operatorname{atan}\left(-\frac{\beta}{\mu_{0}} \frac{\partial A_{z}}{\partial x}\right)$$

For the magnetic fields of interest, it is possible to linearize these expressions to obtain

$$M_{x} = \frac{\chi}{\mu_{0}} \frac{\partial A_{z}}{\partial y}$$

$$M_{y} = -\frac{\chi}{\mu_{0}} \frac{\partial A_{z}}{\partial x}$$
(10-6)

where $\chi = \alpha \beta$ is the magnetic susceptibility.

Boundary Conditions

Along a system boundary reasonably far away from the magnet (see Figure 10-1) you can apply a magnetic insulation boundary condition, $A_z = 0$.

FLUID FLOW EQUATIONS

The Navier-Stokes equations describe the time-dependent mass and momentum balances for an incompressible flow:

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{F}$$

$$\nabla \cdot \mathbf{u} = 0$$
(10-7)

where η denotes the dynamic viscosity (kg/(m·s)), \boldsymbol{u} the velocity (m/s), ρ the fluid density (kg/m³), p the pressure (N/m²), and \boldsymbol{F} a volume force (N/m³).

With the assumption that the magnetic nanoparticles in the fluid do not interact, the magnetic force $\mathbf{F} = (F_x, F_y)$ on the ferrofluid for relatively weak fields is given by $\mathbf{F} = |\mathbf{M}|\nabla|\mathbf{H}|$. Using Equation 10-3, Equation 10-4, and Equation 10-6 then leads to the expressions

$$F_{x} = \frac{\chi}{\mu_{0}\mu_{r}^{2}} \left(\frac{\partial A}{\partial x} z \frac{\partial^{2} A_{z}}{\partial x^{2}} + \frac{\partial A}{\partial y} z \frac{\partial^{2} A_{z}}{\partial x \partial y} \right)$$
$$F_{y} = \frac{\chi}{\mu_{0}\mu_{r}^{2}} \left(\frac{\partial A}{\partial x} z \frac{\partial^{2} A_{z}}{\partial x \partial y} + \frac{\partial A}{\partial y} z \frac{\partial^{2} A_{z}}{\partial y^{2}} \right)$$

To get the final expression for the volume force in the blood stream, multiply these expressions by the ferrofluid mass fraction, $k_{\rm ff}$.

Boundary Conditions

On the vessel walls, apply no-slip conditions, u = v = 0. At the outlet, you can set an outlet pressure condition, p = 0. At the inlet boundary, specify a parabolic flow profile on the normal inflow velocity according to $4U_{m}s(1-s)$, where s is a boundary segment length parameter that goes from 0 to 1 along the inlet boundary segment and U is the maximal flow velocity. To emulate the heart beat, the inflow velocity follows a sinusoidal expression in time:

$$U_0 = 2U_{\rm m}s(1-s)(\sin(\omega t) + \sqrt{\sin^2(\omega t)})$$

Selecting the angular velocity ω to be 2π rad/s gives a heart beat rate of 60 beats per minute. Figure 10-2 displays the resulting expression (normalized to unity).



Figure 10-2: Simulated heart beat.

Model Data

Table 10-1 lists the relevant material properties for the model.

QUANTITY	DESCRIPTION	VALUE
$\mu_{r,mag}$	Relative permeability, magnet	5·10 ³
$B_{\rm rem}$	Remanent flux density, magnet	0.5 T
α	Ferrofluid magnetization-curve parameter	10 ⁻⁴ A/m
β	Ferrofluid magnetization-curve parameter	3·10 ⁻⁵ (A/m) ⁻¹
ρ	Density, blood	1000 kg/m ³
η	Dynamic viscosity, blood	5·10 ⁻³ kg/(m·s)

	TABLE	10-1:	MODEL	DATA
--	-------	-------	-------	------

Results

Figure 10-2 shows a detail from the plot of the magnetic field strength. The highest B-field strength clearly occurs inside the magnet. To see the low-level variations in the surrounding tissue and vessels, the plot does not show magnetic flux densities above 0.17 T. The geometric form of the magnet generates strong fields just outside of the rounded corners. Sharper corners generate even stronger local fields.


Figure 10-3: Magnetic vector potential and magnetic flux density, B field (white areas surpass the plot color range).

Figure 10-4 shows the velocity field at a heart beat where there is a maximum mean throughput in the vessel. At the left end there is a parabolic laminar flow profile.



Figure 10-4: Velocity field at maximum blood throughput (t = 0.25).

Figure 10-5 reveals the velocity field between two heart beats, where the net throughput is zero.



Figure 10-5: Velocity field at zero blood throughput (t = 1).

Modeling in COMSOL Multiphysics

EQUATIONS

To model Equation 10-5, use the Magnetostatics application mode in 2D. In the **Subdomain Settings** dialog box, choose the constitutive relation

$$\mathbf{B} = \mu_0 \mu_r \mathbf{H} + \mathbf{B}_r$$

and make use of the fact that the remanent flux density $\mathbf{B}_{r} = \mu_{0} \chi \mathbf{M}$, where χ is the magnetic susceptibility.

STAGED SOLUTION

Because the magnetostatic problem is a stationary nonlinear problem that is independent of the fluid-flow problem, you only need to solve that once. You can therefore start by solving only Equation 10-5 with the stationary solver. Then proceed with solving only the fluid-flow problem, Equation 10-7, with the static magnetic

potential as input. Solve the fluid-flow problem using the time-dependent solver. This strategy reduces RAM memory allocation and speeds up the solution.

SOLVER SETTINGS

The convergence tolerance for the time-stepping algorithm should only be based on the truly time-dependent equations. The continuity equation (second line in Equation 10-7) is stationary and describes the pressure distribution. Therefore, you can exclude p from the time-stepping tolerance checks. To do so, select **Exclude** algebraic in the **Error estimation strategy** list on the **Time Stepping** page of the **Solver Parameters** dialog box.

FINITE ELEMENT SHAPE FUNCTIONS

This problem includes second-order space derivatives of *A* in some coefficients. To get acceptable accuracy, use third-order Lagrange elements in the Magnetostatics application mode. The default setting in this application mode is second-order elements.

References

1. P.A. Voltairas, D.I. Fotiadis, and L.K. Michalis, "Hydrodynamics of Magnetic Drug Targeting," *J. Biomech.*, vol. 35, pp. 813–821, 2002.

2. C.M. Oldenburg, S.E. Borglin, and G.J. Moridis, "Numerical Simulation of Ferrofluid Flow for Subsurface Environmental Engineering Applications," *Transport in Porous Media*, vol. 38, pp. 319–344, 2000.

3. R.E. Rosensweig, Ferrohydrodynamics, Dover Publications, New York, 1997.

Model Library path: COMSOL_Multiphysics/Multiphysics/ magnetic_drug_targeting

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Start COMSOL Multiphysics.
- 2 In the Model Navigator, select 2D from the Space dimension list.

- 3 Select COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis.
- 4 Click first the Multiphysics button and then the Add button.
- 5 From the list of application modes, select COMSOL

Multiphysics>Electromagnetics>Magnetostatics>Stationary analysis. Click Add.

6 Click OK.

GEOMETRY MODELING

- I Click the **Rectangle/Square** button on the top of the Draw toolbar. Draw an arbitrary rectangle by clicking in the drawing area and dragging the mouse.
- **2** Double-click the rectangle.
- 3 Make the following changes in the dialog box; when done, click **OK**.

PROPERTY	VALUE
Width	0.02
Height	0.06
Base x	-0.05
Base y	0

- 4 Click the **Zoom Extents** button on the Main toolbar.
- **5** From the **Draw** menu, choose **Fillet/Chamfer**. Click on the rectangle R1 and select all four vertices by Ctrl-clicking them.
- 6 Type 1e-3 in the Radius edit field, then click OK.
- 7 Draw three additional arbitrary rectangles using the **Rectangle/Square** button and set the object properties like the first one according to the following table:

RECTANGLE	WIDTH	HEIGHT	BASE X	BASE Y
R2	0.22	0.05	-0.14	-0.07
R3	0.22	0.02	-0.14	-0.02
R4	0.22	0.13	-0.14	-0.01



8 Click the Zoom Extents button on the Main toolbar.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- 2 Enter constants with names, expressions (values and units), and descriptions (the descriptions are optional) according to the following table; when done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
mur_mag	1	Relative permeability, magnet
B_rem	1[T]	Remanent flux density, magnet
alpha	1e4[A/m]	Ferrofluid magnetization-curve parameter
beta	3e-5[m/A]	Ferrofluid magnetization-curve parameter
chi_ff	alpha*beta	Magnetic susceptibility, ferrofluid
k_ff	0.1	Ferrofluid mass fraction in blood stream
rho	1000[kg/m^3]	Density, blood
eta	0.005[Pa*s]	Dynamic viscosity, blood
U_m	50[cm/s]	Maximum flow velocity
f	60[1/min]	Heart-beat rate
omega	2*pi[rad]*f	Pulse angular velocity

Scalar Expressions

I On the **Options** menu, point to **Expressions** and then click **Scalar Expressions**.

NAME	EXPRESSION	DESCRIPTION
muO	mu0_qa	Permeability of vacuum
M_ffx	k_ff*(chi_ff/mu0)*Azy	Induced ferrofluid magnetization, x-component
M_ffy	-k_ff*(chi_ff/mu0)*Azx	Induced ferrofluid magnetization, y-component
F_ffx	k_ff*(Azx*Azxx+Azy*Azxy)* chi_ff/(muO*(1+chi_ff)^2)	Ferrofluid volume force, x-component
F_ffy	k_ff*(Azx*Azxy+Azy*Azyy)* chi_ff/(muO*(1+chi_ff)^2)	Ferrofluid volume force, y-component

2 Define the following expressions (the descriptions are optional):

3 Click OK.

PHYSICS SETTINGS

Application Mode Properties

Change the default element type to be a cubic Lagrange element:

- I From the Multiphysics menu, choose Magnetostatics (qa).
- 2 From the Physics menu, choose Properties.
- **3** In the **Application Mode Properties** dialog box, select **Lagrange Cubic** in the **Default element type** list. Click **OK**.

Subdomain Settings—Magnetostatics

- I From the Physics menu, choose Subdomain Settings.
- 2 On the Physics page, select Subdomain 2.
- **3** Set the **Constitutive relation** to $\mathbf{B} = \mu_0 \mathbf{H} + \mu_0 \mathbf{M}$.
- 4 In the left and right M edit fields, enter M_ffx and M_ffy, respectively.
- **5** Select Subdomain 5, then set the **Constitutive relation** to $\mathbf{B} = \mu_0 \mu_r \mathbf{H} + \mathbf{B}_r$.
- 6 In the μ_r edit field, enter mur_mag. In the right B_r edit field for the y-component of the remanent flux density, enter B_rem. Leave the corresponding x-component at zero.
- 7 Click OK.

Boundary Conditions-Magnetostatics

The default setting is magnetic insulation, so you do not have to make any changes. To be reassured, choose **Physics>Boundary Settings** and inspect the boundaries.

Subdomain Settings—Navier-Stokes

- I From the Multiphysics menu, choose Incompressible Navier-Stokes (ns).
- 2 From the Physics menu, choose Subdomain Settings.
- **3** Press Ctrl+A to select all five subdomains.
- 4 Clear the Active in this domain check box.
- 5 Select Subdomain 2 from the list and select the Active in this domain check box.
- **6** Specify the fluid properties according to the following table:

QUANTITY	VALUE/EXPRESSION
ρ	rho
η	eta
F _x	F_ffx
Fy	F_ffy

- **7** Click the **Artificial Diffusion** button to add streamline diffusion to the model. This is necessary because of the high Reynolds number flow.
- 8 In the Artificial Diffusion dialog box, select the Streamline diffusion check box and then click OK.
- 9 On the Init page, leave all initial conditions at zero.

IO Click OK.

Boundary Conditions-Navier-Stokes

I From the Physics menu, choose Boundary Settings.

2 Specify boundary conditions according to the following table:

SETTINGS	BOUNDARY 3	BOUNDARIES 4, 6	BOUNDARY 16
Boundary type	Inlet	Wall	Outlet
Boundary condition	Velocity	No slip	Pressure, no viscous stress
U ₀	2*s*(1-s)*U*(sin(omega*t)+ sqrt(sin(omega*t)^2))		
Po			0

3 Click OK.

MESH GENERATION

- I Choose Mesh>Free Mesh Parameters, and in the dialog box that opens, click the Subdomain tab.
- 2 Select Subdomain 2 from the list and enter 0.001 in the Maximum element size edit field. Click Remesh and OK.



COMPUTING THE SOLUTION

- I Click the Solver Manager button on the Main toolbar.
- 2 On the Solve For page, select only Magnetostatics (qa). Click OK.
- 3 Click the Solver Parameters button on the Main toolbar.
- 4 On the General page, select Stationary from the Analysis list. Click OK.
- **5** Click the **Solve** button on the Main toolbar to compute the solution for A_2 .
- 6 When the solver has finished, click the Solver Manager button.
- 7 On the Initial Value page, go to the Values of variables not solved for and linearization point area and click the Current solution option button.
- 8 On the Solve For page, select only Incompressible Navier-Stokes (ns). Click OK.
- 9 Click the Solver Parameters button on the Main toolbar.

- 10 On the General page, select Transient from the Analysis list and type 0:0.05:2 in the Times edit field.
- II On the **Time Stepping** page, select **Exclude algebraic** from the **Error estimation strategy** list (see "Solver Settings" on page 350 for an explanation). Click **OK**.
- 12 Click the Solve button on the Main toolbar.

The solution takes a few minutes to compute.

POSTPROCESSING AND VISUALIZATION

To generate Figure 10-3 on page 347, follow these instructions.

- I Click the Plot Parameters button on the Main toolbar.
- **2** On the **General** page, select the **Surface** and **Contour** and check boxes in the **Plot type** area.
- 3 Click the Surface tab. On the Surface Data page, select Magnetostatics (qa)>Magnetic flux density, norm from the Predefined quantities list.
- 4 On the Contour page, select Magnetostatics (qa)>Magnetic potential, z component.
- 5 Click OK.

To generate Figure 10-4 on page 348 and Figure 10-5 on page 349, proceed in the following way:

- I Click the Plot Parameters button on the Main toolbar.
- 2 On the General page, select 0.25 from the Solution at time list.
- 3 Click the Surface tab. On the Surface Data page, select Incompressible Navier-Stokes (ns)>Velocity field from the Predefined quantities list.
- **4** Click **Apply** to generate the plot in Figure 10-4.
- 5 On the General page, change the entry in the Solution at time list to I.
- 6 Click **OK** to generate the plot in Figure 10-5.

To generate a movie, do as follows:

- I Open the **Plot Parameters** dialog box again and go to the **Animate** tab.
- 2 Click **OK** and wait for the movie to be generated (this step can take some time). When the movie is finished, you can save it as an AVI or QuickTime file using the **Save** button in the movie window.

Free Convection

Introduction

This example describes a fluid flow problem with heat transfer in the fluid. An array of heating tubes is submerged in a vessel with fluid flow entering at the bottom. Figure 10-6 shows the setup.



Figure 10-6: Heating tubes and direction of the fluid flow

Model Definition

The first consideration when modeling should always be the true dimension of the problem. Sometimes there are no variations in the third dimensions, and it can be extrapolated from the solution of a related 2D case. Neglecting any end effects from

the walls of the vessel, the solution is constant in the direction of the heating tubes, therefore you can reduce the model to a 2D domain.



Figure 10-7: Using symmetry to reduce computation time and complexity. The model describes one section of the array of heating tubes (indicated by the dashed lines).

The next step is finding symmetries. In this case, using symmetry planes, it suffices to model the thin domain indicated in Figure 10-7.

GOVERNING EQUATIONS

This is a multiphysics model because it involves more than one kind of physics. The incompressible Navier-Stokes equations from fluid dynamics work together with a heat transfer equation. There are four unknown field variables (dependent variables):

- The velocity field components, *u* and *v*
- The pressure, p
- The temperature, T

They are all related through bidirectional multiphysics couplings.

The incompressible Navier-Stokes equations consist of a momentum balance (a vector equation) and a mass conservation and incompressibility condition. The equations are:

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \eta \nabla^2 \mathbf{u} + \mathbf{F} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

with the following variables:

- **u** is the velocity field.
- *p* is the pressure.
- **F** is a volume force.
- ρ is the fluid density.
- η is the dynamic viscosity.
- ∇ is the vector differential operator.

The heat equation is an energy conservation equation that says that the change in energy is equal to the heat source minus the divergence of the diffusive heat flux:

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) + \nabla \cdot (-k \nabla T) = Q$$

where c_p is the heat capacity of the fluid, and ρ is fluid density. Q represents a source term. The velocity field comes from the incompressible Navier-Stokes equation.

Modeling in COMSOL Multiphysics

To build a model in COMSOL Multiphysics using the above equations, use two application modes: the Incompressible Navier-Stokes application mode for fluid flow and the Convection and Conduction application mode for heat transfer. The multiphysics couplings enter directly into the physics settings in the application modes.

In this model, the equations are coupled in both directions. First, add free convection to the momentum balance with the *Boussinesq approximation*. This approximation ignores variations in density with temperature, except that the variations give rise to a buoyancy force lifting the fluid. This force enters the \mathbf{F} term in the incompressible Navier-Stokes equations.

At the same time, the heat equation must account for the velocity field. Using the predefined Fluid-Thermal Incompressible Flow coupled multiphysics entry, the velocity components from the incompressible Navier-Stokes equations appear

automatically as the velocity field for the convective heat transfer in the Convection and Conduction application mode.

Results

The analysis of the coupled thermal-fluid model provides the velocity field, pressure distribution, and temperature distribution in the fluid. Figure 10-8 shows a postprocessing plot of the velocity field and the temperature. Without heating, you would expect an exit *y*-velocity that is slightly lower toward the left side, behind the heating tube (wake effect). In this case, however, you see that the *y*-velocity is higher on the left side. This is because of the buoyancy effect of the free convection.



Figure 10-8: The velocity field and temperature distribution in the fluid.

Using integration to find the mean temperature at the outlet shows that the temperature increases roughly 0.75 K from the inlet to the outlet.

For further analysis, see "Time-Dependent Simulation" on page 373.

Model Library path: COMSOL Multiphysics/Multiphysics/free_convection

MODEL NAVIGATOR

Use the following steps to create a new multiphysics model:

- I Click the **New** tab, and check that **2D** is selected in the **Space dimension** list. The space dimension must always be selected first, because the available application modes vary depending on the space dimension.
- 2 In the application modes list on the left, open the **COMSOL Multiphysics** folder, open the **Fluid-Thermal Interaction** folder, and then click **Fluid-Thermal Incompressible Flow**.
- 3 Click OK to close the Model Navigator and create a new model.



Figure 10-9: Selecting the Fluid-Thermal Incompressible Flow predefined multiphysics coupling in the Model Navigator.

OPTIONS AND SETTINGS

Later in this model you need the fluid properties of the water, the temperatures at the inlet and on the surface of the heating tubes, and the inlet velocity. It is convenient to enter this data as *constants* in the **Constants** dialog box. In this model, all values are given in SI units.

- I From the **Options** menu, choose **Constants**.
- 2 First add the density of the fluid:

Enter rho0 in the Name field. Press the Tab key to move the cursor to the Expression field and enter 1e3[kg/m^3]. The value is saved when you press Enter, click Apply,

or otherwise leave the **Expression** field. In the **Description** column, type Fluid density (adding a description is optional).

NAME	EXPRESSION	DESCRIPTION
rho0	1e3[kg/m^3]	Fluid density
mu	1e-3[Pa*s]	Dynamic viscosity
Ср	4.2e3[J/(kg*K)]	Heat capacity
kc	0.6[W/(m*K)]	Thermal conductivity
alpha0	0.18e-3[1/K]	Thermal expansion coefficient
g0	9.8[m/s^2]	Acceleration due to gravity
vin	5e-3[m/s]	Inlet velocity
Tin	293[K]	Inlet temperature
Theat	303[K]	Heater temperature

3 Continue by adding the remaining properties

4 When you have entered all constants, click OK.

rho0 [1e3[kg/m^3] 1000[kg/m ³] Fluid density mu [1e-3[Pa*s] 0.001[kg/(m·s)]Dynamic viscosity	1
mu 1e-3[Pa*s] 0.001[kg/(m·s)]Dynamic viscosity	-
	1
Cp [4.2e3[J/(kg*K)] [4200[J/(kg•K)] Heat capacity	
kc 0.6[W/(m*K)] 0.6[W/(m-K)] Thermal conductivity	
alpha0 0.18e-3[1/K] (1.8e-4)[1/K] Thermal expansion coe	fficient
g0 9.8[m/s^2] 9.8[m/s ²] Acceleration due to gra	vity
vin 5e-3[m/s] 0.005[m/s] Inlet velocity	
Tin 293[K] 293[K] Inlet tempertature	
Theat 303[K] 303[K] Heater temperature	

The model size is in the order of a few centimeters, while the area visible in the graphical interface is in the order of meters (remember that you are modeling in SI units). Before you start drawing the geometry, change the size of the visible drawing area and the grid spacing:

I From Options menu, choose Axes/Grid Settings.

2 In the Axes/Grid Settings dialog box, type -0.01, 0.01, -0.01, and 0.05 in the x min, x max, y min, and y max edit fields, respectively.

Axes/Grid	Settings	i		X
Axis Gri	o equal			
-x-y limit	s		z limits	
x min:	-0.01		Auto	
x max:	0.01		z min: -1	
y min:	-0.01		z max: 1	
y max:	0.05			
		ОК	Cancel Apply Help	

- **3** Click the **Grid** tab and then click to clear the **Auto** check box. Enter 0.005 in both the **x spacing** and **y spacing** edit fields.
- 4 Click **OK** to close the dialog box and apply the settings.

GEOMETRY MODELING

The next step is to create the model geometry. This is easy, involving only a rectangle and circle geometry object, subtracting one from the other.

- I Draw a rectangle of width 0.005 and height 0.04, with the lower left corner at the origin. Click the **Rectangle/Square** toolbar button. It is the first button on the Draw toolbar, on the left side of the drawing area. Then click at (0,0) using the left mouse button, and drag the mouse to (0.005,0.04). Release the button.
- 2 Draw a circle with radius 0.005 centered at (0, 0.015). Click the fourth button on the Draw toolbar, **Ellipse/Circle (Centered)**. Then, using the *right* mouse button, click at (0, 0.015) and drag the mouse in any direction, keeping the button down, until the circle has a radius of 0.005. Using the right mouse button constrains the ellipse to a circle.
- **3** The desired radius of the circle is not 0.005, however, but is 0.0025. To fix this, double-click the circle object or select **Object Properties** from the **Draw** menu. In the **Circle** dialog box, enter 0.0025 as **Radius** and click **OK**.

Circle			X
Size Radius:	0.0050	Rotal a:	tion angle 0 (degrees)
Position			
Base:	Center 👻	Style:	Solid 👻
x	0	Name:	C1
y:	0.015		
	OK Cancel		Apply Help

4 Select both the circle and the rectangle. Either draw a rubber-band box around both them, or press the shortcut key Ctrl+A to select all objects.



5 Click the Difference button on the Draw toolbar and then the Zoom Extents button on the Main toolbar. The Zoom Extents button adjusts the axes settings so that the geometry fits nicely into the drawing area.



PHYSICS SETTINGS

Boundary Conditions

Specify the boundary conditions, first for the Convection and Conduction application mode and then for the Incompressible Navier-Stokes application mode.

- I From the Physics menu, choose Boundary Settings. This opens the Boundary Setting dialog box and transfers COMSOL Multiphysics to the *boundary selection mode*. The contents of this dialog box vary depending on the application mode.
- **2** Start by insulating all boundaries. This is the default boundary condition, so you do not need to change anything.
- **3** Select the inflow boundary by clicking at the corresponding edge in the geometry or by selecting Boundary 2 from the **Boundary selection** list. Select **Temperature** from the **Boundary conditions** list and type **Tin** in the **Temperature** edit field.

4 Select the heater that exists along Boundaries 6 and 7 from the **Boundary selection** list. Then select **Temperature** in the **Boundary conditions** list and type Theat in the **Temperature** edit field.

Boundary Settings - Convecti	on and Conduction (c	c)		8
Boundary Settings - Convecti Equation T = T ₀ Boundaries Groups Boundary selection 1 2 3 4 5 6 7 •	on and Conduction (c Coefficients Color/St Boundary conditions Boundary condition: Quantity q ₀ T ₀	c) <u>(Temperature</u>) • Value/Expression 0 Theat	Unit Description W/m ² Inward heat Flu K Temperature	x
Select by group				
	0	K Cancel	Apply Help)

Figure 10-10: Specifying a temperature of Theat on boundaries 6 and 7.

- **5** Click the outflow boundary (Boundary 4) and select **Convective flux** in the **Boundary conditions** list.
- **6** Click **Apply** to confirm the settings for the Convection and Conduction application mode.
- 7 Switch to the Incompressible Navier-Stokes application mode by selecting this mode from the Multiphysics menu. You can now open the corresponding Boundary Settings dialog box from the Physics menu (you can also use the Model Tree to access the settings for both applications modes).
- 8 Select any boundary in the **Boundary selection** list and then press Ctrl+A to select all boundaries in the list and then select the **Symmetry boundary** from the **Boundary type** list.

9 For the inflow boundary, select Boundary 2 and then choose Inlet from the Boundary type list and Velocity from the Boundary condition list. Type vin in the Normal inflow velocity edit field.

Boundary Settings - Incompr	essible Navier-Stokes (ns)	ר
Equation		
$\mathbf{u} = - \cup_0 \mathbf{n}$		
Boundaries Groups	Coefficients Color/Style	
Boundary selection	Boundary conditions	
1	Boundary type: Inlet	
3	Boundary condition: Velocity	
4	Quantity Value/Expression Unit Description	
5	O m/s x-velocity	
7 +	v ₀ 0 m/s y-velocity	
Group: 🚽	U0 vin m/s Normal inflow velocity	
Select by group		
Interior boundaries		
	OK Cancel Apply Help	

Figure 10-11: Specifying the velocity in the normal direction.

- **10** Continue by selecting the outflow boundary (Boundary 4). From the **Boundary type** list, select **Outlet**. The default condition **Pressure**, **no viscous stress** is correct.
- II Finally select Boundaries 6 and 7 and then select the type **Wall** and the condition **No slip**. Click **OK** to confirm your choices and close the dialog box.

Subdomain Settings

I From the **Physics** menu, choose **Subdomain Settings** to open the **Subdomain Settings** dialog box and put the user interface in the Subdomain Selection mode.

2 Select the single subdomain, Subdomain 1, and use the already defined constants to set the density and the viscosity. Type rho0 and mu in the **Density** and **Dynamic** viscosity fields, respectively.

Subdomain Settings - Incom	pressible Navier-Stokes (ns)
Equations	
$\rho \mathbf{u} \cdot \nabla \mathbf{u} = \nabla \cdot [-\rho \mathbf{I} + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u} + \nabla \mathbf{u}))]$	$\nabla \mathbf{u}^{T}$] + F
$\nabla \cdot \mathbf{u} = 0$	
Subdomains Groups	Physics Init Element Color,
Subdomain selection	Fluid properties and sources/sinks
1 (default)	Library material:
Gran: default	Quantity Value/Expression Unit Description p rho_ccc kg/m³ Density n mu Pa-ss Dynamic viscosity Fg 0 N/m³ Volume force, x-dir. Fy alpha0*g0*rho0*(T- N/m³ Volume force, y-dir.
Select by group	Artificial Diffusion
	OK Cancel Apply Help

Figure 10-12: Providing fluid properties and a volume force in the Subdomain Settings dialog box.

3 To model the effect of temperature on the density of the fluid as a buoyancy force in the y direction, type alpha0*g0*rho0*(T-Tin) in the **Volume force**, y dir. edit field (F_y) . Remember that T is the dependent variable from the Convection and Conduction application mode. Notice that an extended edit field displays the entire expression.

In the **Subdomain Settings** dialog box you can also set initial values. The Navier-Stokes equations are nonlinear and therefore require an educated guess as an initial solution to the nonlinear solver. A good initial value is necessary for the model to converge. The initial values also serve as initial conditions for the time-dependent solver. In this case, a constant velocity field v_{in} has sufficient quality.

I Click the **Init** tab in the **Subdomain Settings** dialog box. Set the initial value $v(t_0)$ to vin.

Subdomain Settings - Incom	pressible Navier-Stokes (ns)	X
Equations		
$\rho \mathbf{u} \cdot \nabla \mathbf{u} = \nabla \cdot [-\rho \mathbf{I} + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u} + \nabla \mathbf{u}))]$	7u) ^T)] + F	
$\nabla \cdot \mathbf{u} = 0$		
Subdomains Groups	Physics Init Element Color	
Subdomain selection	Initial value	
1 (default)	u(t ₀) 0	m/s x-velocity
	v(t ₀) vin	m/s y-velocity
	p(t ₀) 0	Pa Pressure
-		
Group: default -		
Select by group		
Active in this domain		
	OK Cancel	Apply Help

Figure 10-13: The initial values for the velocity components.

2 Click OK.

Continue with the subdomain settings for the Convection and Conduction application mode:

I Use the **Multiphysics** menu to switch to the Convection and Conduction application mode.

2 From the **Physics** menu, choose **Subdomain Settings** to open the **Subdomain Settings** dialog box for the Convection and Conduction application mode.

Subdomain Settings - Convect	tion and Conduction	(cc)		X
Equation				
$\nabla \cdot (-k \nabla T) = Q - \rho C_p \mathbf{u} \cdot \nabla T$				
T = temperature				
Subdomains Groups	Physics Init Elemen	nt Color		
Subdomain selection	Thermal properties a	and heat sources/sinks		
1 (default)	Library material:	•	Load	
	Quantity	Value/Expression	Unit	Description
	k (isotropic)	kc	W/(m•K)	Thermal conductivity
	💿 k (anisotropic)	0.025 0 0 0.025	W/(m∙K)	Thermal conductivity
	ρ	rho0	kg/m ³	Density
	C,	Ср	J/(kg-K)	Heat capacity at constant pressure
	Ŷ	1		Ratio of specific heats
Group: default 👻	Q	0	W/m ³	Heat source
Select by group	u	u v	m/s	Velocity field
Active in this domain	Artificial Diffusio	n		
		Ok		Cancel Apply Help

Figure 10-14: Specifying the thermal properties.

- **3** Type rho0, Cp, and kc in the **Density**, **Heat capacity**, and **Thermal conductivity** (isotropic) edit fields, respectively.
- 4 Notice that u and v appear in the x-velocity and y-velocity edit fields. They are the dependent variables for the velocity components from the Incompressible Navier-Stokes application mode, which account for the convective transport of heat. This is the predefined multiphysics coupling that the Fluid-Thermal Incompressible Flow coupled multiphysics entry sets up automatically.

5 The temperature also needs an initial value. Use the inlet temperature as the starting temperature in the entire domain. Click the lnit tab and enter Tin in the T(t₀) field. Then click OK to confirm all settings and close the dialog box.

ubdomain Settings - Convection and Conduction (cc)	X
Equation	
$\nabla \cdot (-k \nabla T) = Q - \rho C_{p} \mathbf{u} \cdot \nabla T$	
T = temperature	
Subdomains Groups Physics Init Element Color	_
Subdomain selection Initial value	
1 (default) T(t ₀) Tin K Temperature	
Group: default Group: default Group: default Group Group Group Group Group	
OK Cancel Apply Help	>

Figure 10-15: The initial value for the temperature.

MESH GENERATION

In this model, use a predefined setting for a fine mesh:

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 In the Free Mesh Parameters dialog box, select Finer in the Predefined mesh sizes list.
- 3 Click OK.
- 4 Initialize the default mesh clicking the **Initialize Mesh** button on the Main toolbar.

Later you can refine the mesh to further improve the accuracy of the solution. If solving again using a finer mesh gives similar results, the mesh resolution is sufficient. Otherwise, you must refine the mesh again.

COMPUTING THE SOLUTION

The required nonlinear solver is the default solver, so you start the solver directly.

From the **Solve** menu choose **Solve Problem**, or click the **Solve** button on the Main toolbar. Notice the solution progress window where you can monitor and stop the solution process. The solution converges quickly in this case.

As soon as the solution is ready, COMSOL Multiphysics displays a default plot. In this case, you get a surface plot of the velocity field.

POSTPROCESSING AND VISUALIZATION

In *postprocessing mode* you can select from many different plot types and set parameters for the different plots. Using the postprocessing utilities you can visualize the solution variables, their derivatives, and the space coordinates. Many frequently used expressions are predefined as *application mode variables*, directly available from lists in the **Plot Parameters** dialog box. Perform the following steps to create the plot in Figure 10-8:

- I Click the **Plot Parameters** button on the Main toolbar. This opens the **Plot Parameters** dialog box.
- 2 To obtain a surface plot with superimposed velocity vectors, select the **Surface** and **Arrow** check boxes in the **Plot type** area on the **General** tab.
- **3** Click the **Surface** tab.
- 4 On the Surface Data tab, select Convection and Conduction (cc)>Temperature from the Predefined quantities list.
- **5** Click the **Arrow** tab.
- 6 Select Incompressible Navier-Stokes (ns)>Velocity field from the Predefined quantities list on the Subdomain Data tab.
- **7** In the **Arrow positioning** area, type 10 and 15 in the **x points** and **y points** edit fields, respectively.
- 8 Finally, in the **Arrow parameters** area, click the **Color** button to choose a suitable arrow color, for example, white. When you have selected a color, click **OK** to close the **Arrow Color** dialog box.
- 9 Click OK.

ADVANCED POSTPROCESSING

Integrating to Find the Mean Temperature

Because both the velocity and temperature vary along the outlet at the top, you must use additional postprocessing to get the mean temperature at the exit. The bulk temperature, or the "cup mixing temperature" is the temperature that the fluid has if it is collected in a cup at the outflow and is properly mixed. For this 2D example, assuming constant c_p and ρ , the cup mixing temperature is given by the following expression on boundary 4 (the outlet):

$$\langle T \rangle = \frac{\int T(\mathbf{n} \cdot \mathbf{u}) ds}{\int (\mathbf{n} \cdot \mathbf{u}) ds}$$

where \mathbf{u} is the velocity vector and the \mathbf{n} is the unit normal vector of the boundary.

- I To obtain the denominator, choose **Boundary Integration** from the **Postprocessing** menu.
- 2 Select Boundary 4.
- **3** Type nx*u+ny*v in the **Expression** edit field.
- 4 Click Apply.
- **5** The message log at the bottom of the main user interface shows the value of the denominator:

Value of integral: 2.5e-5 [m^2/s], Expression: nx*u+ny*v, Boundary: 4.

- 6 In the Boundary Integration dialog box, type T*(nx*u+ny*v)/2.5e-5 in the Expression edit field. This computes the resulting mean temperature.
- 7 Click **OK**. The result is displayed in the message log.

You should get a mean temperature of roughly 293.8 K, that is, a temperature rise of approximately 0.8 K between inlet and outlet.

Time-Dependent Simulation

This example is an extension of the stationary free convection model from the previous section. In the first step of modeling the heat transfer by forced and free convection, we found a steady-state solution. See "Modeling Using the Graphical User Interface" on page 361 on how to create and save the first part of the model.

Modeling Using the Graphical User Interface

The following section assumes that you start from the final state of the steady-state model.

LOADING A PREVIOUSLY SAVED MODEL

If you have previously saved the results from the first part of the model as a Model MPH-file, now load it into COMSOL Multiphysics (if you are continuing from the previous steady-state model, skip the following three steps):

- I From the File menu, choose **Open**. You can also click the **Open** button on the main toolbar to load a Model MPH-file.
- 2 Browse to locate the Model MPH-file and select it.
- 3 Click Open.

SELECTING TRANSIENT ANALYSIS

Before solving the model with the time dependent solver, we need to set the analysis type to transient for both application modes:

- I For the current application mode (Convection and Conduction), select **Properties** from the **Physics** menu.
- 2 In the dialog that appears, set Analysis type to Transient and click OK.
- 3 Select Incompressible Navier-Stokes (ns) from the Multiphysics menu.
- 4 Select Properties from the Physics menu.
- 5 In the dialog that appears, set Analysis type to Transient and click OK.

COMPUTING THE SOLUTION

- I Click the General tab in the Solver Parameters dialog box.
- **2** The **General** page for the time-stepping algorithms is now active. The output times can be a vector, specifying for which times the solution should be saved for

postprocessing. Type 0:1:30 in the **Times** edit field to specify that the solution will be sampled every second during 30 seconds.

nalysis:	General	Time Stepping	Advanced				
Transient	-	Tanta arappang					
Auto select solver	lime s	tepping					-
ahuaru	Times:			0:1:30 0.01 0.0010			_
olver:	Relativ	ve tolerance:					-
ime dependent	Absolu	ite tolerance:					
igenvalue		low complex nu	mbers				
arametric							
tationary segregated	Linear	system solver					
Parametric segregated	Linear	system solver:	Direct (PARDIS	D)	-		
	Precor	nditioner:	-		÷		
	1000000						
	-						
	-				Ē	C-11/	
Adaptive mesh reinemen	nu -					Settings	
Optimization	Makeine	europa a bro u	Automotic		1		
	Macrix	symmetry:	Automatic		▼.		
	à.						

Figure 10-16: Setting the parameters for the time-dependent analysis.

- **3** Click **OK** to confirm the choices.
- 4 Click the **Solve** button on the Main toolbar to start the time-dependent analysis.

POSTPROCESSING AND VISUALIZATION

The plot shows the solution for the last time step using the same plot settings as before. Select which time step to display in the **Plot Parameters** dialog box:

- I Click the **Plot Parameters** button to open the **Plot Parameters** dialog box.
- 2 Click the General tab and select the time step that you want to see from the Solution at time list.



3 Click **Apply** to plot the solution at that time.

Marangoni Convection

Introduction

Marangoni convection occurs when the surface tension of an interface (generally liquid-air) depends on the concentration of a species or on the temperature distribution. In the case of temperature dependence, the Marangoni effect is also called thermo-capillary convection. It is of primary importance in the fields of:

- Welding
- Crystal growth
- Electron beam melting of metals

Direct experimental studies are not easy to carry out in these systems because the materials are often metals and temperatures are very high. One possibility is to replace the real system with an experimental setup using a transparent liquid at ambient temperatures.

Model Definition

This model describes the 2D stationary behavior of a vessel filled with silicone oil, for which the thermo-physical properties are known. The aim of the study is to compute the temperature field that induces a flow through the Marangoni effect. The model shows this effect using the simple geometry in the figure below.



GOVERNING EQUATIONS

A stationary momentum balance describes the velocity field and the pressure distribution (Navier-Stokes equations):

$$-\eta \nabla^2 \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{F}$$
$$\nabla \cdot \mathbf{u} = 0$$

The first term gives the rate of momentum gain by viscous transfer, the second by convection, and the third by pressure forces. The equation contains the following variables and properties:

- η is the dynamic viscosity (kg/(m·s)).
- **u** is the velocity vector (m/s).
- ρ is the density (kg/m³)
- *p* is the pressure (Pa).

The **F** term is a source term representing external forces per unit volume.

To include the heating of the fluid, couple the fluid flow to an energy balance according to

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

The expression within the brackets is the heat flux vector, containing a conductive and a convective part. The equation contains the following variables and properties:

- k is the thermal conductivity (W/(m·K)).
- *T* is the temperature (K).
- C_p is the heat capacity $(J/(kg \cdot K))$.
- $Q (W/m^3)$ represents a source term.

You can use the Boussinesq approximation to include the effect of temperature on the velocity field. In this approximation, variations in temperature produce a buoyancy force (or Archimedes' force) that lifts the fluid. Enter this force into the \mathbf{F} term in the Navier-Stokes equation as:

$$\mathbf{F} = \begin{pmatrix} Fx \\ Fy \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha g \rho \left(T - \frac{\Delta T}{2} \right) \end{pmatrix}$$

where α is the thermal expansion coefficient (1/K), *g* is the acceleration due to gravity (m/s²), and ΔT is the temperature difference (K) between the right and left wall in the model geometry.

Equation 10-8 describes the forces that the Marangoni effect induces on the interface (liquid/air):

$$\eta \frac{\partial u}{\partial y} = \gamma \frac{\partial T}{\partial x} \tag{10-8}$$

where γ is the temperature derivative of the surface tension (N/(m·K)). This equation states that the shear stress on a surface is proportional to the temperature gradient (Ref. 1).

Modeling in COMSOL Multiphysics

To solve the momentum balance equations, use the Incompressible Navier-Stokes application mode. For the heat transfer by convection and conduction, use the Convection and Conduction application mode. There are two multiphysics couplings, one in each direction:

- The Boussinesq approximation means that an expression including temperature acts as a force in the *y* direction in the momentum balance.
- The convective heat transfer depends on the velocities from the momentum balance.

This means that you must solve the coupled system directly using the nonlinear solver.

To implement the condition that the shear stress is proportional to the temperature gradient on the surface, use a Weak Form, Boundary application mode where you can implement Equation using a weak formulation:

```
lm_test*(eta1*uy-gamma*Tx)+u_test*lm
```

The last term makes sure that this condition replaces the condition on the *x*-velocity from the Slip/Symmetry boundary condition in the Incompressible Navier-Stokes application mode.

It is also necessary to omit the Dirichlet boundary condition for T at the surface level to get consistent boundary conditions. To do this, adjust the boundary conditions on the equation system level.

Finally, to study the Marangoni effect, increase the temperature difference between the right and left wall in steps. With the restart option, the solver uses the previous solution as initial value providing fast convergence.

Results





Figure 10-17: Marangoni convection with a temperature difference of 0.001 K.

For the very low temperature difference of 0.001 K, the temperature field is almost decoupled from the velocity field. Therefore, the temperature decreases almost linearly from left to right.



Figure 10-18: Marangoni convection with a temperature difference of 0.05 K.

For the temperature difference of 0.05 K notice how the Marangoni convection influences the flow of fluid and the distribution of temperature. The temperature is no longer decreasing linearly and you can clearly see the advection of the isotherms caused by the flow.



Figure 10-19: Marangoni convection with a temperature difference of 2 K.

At higher temperature differences (2 K in Figure 10-19 above), the physical coupling between the temperature and the velocity field is clearly visible. The heat conduction is small compared to the convection, and at the surface the fluid accelerates where the temperature gradient is high.

Reference

1. V.G. Levich, Physicochemical Hydrodynamics, Prentice-Hall, N.J., 1962.

Model Library path: COMSOL_Multiphysics/Multiphysics/marangoni

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

I In the Model Navigator, select 2D from the Space dimension list.
- 2 Click the **Multiphysics** button.
- 3 In the COMSOL Multiphysics>Fluid Dynamics folder, select Incompressible Navier-Stokes and then Steady-state analysis. Make sure Lagrange - P₂P₁ is selected in the Element list.
- 4 Click Add.
- 5 In the Heat Transfer folder, select Convection and Conduction and then Steady-state analysis. Make sure Lagrange Quadratic is selected in the Element list.
- 6 Click Add.
- 7 In the PDE Modes folder, select Weak Form, Boundary and then Stationary analysis.
- 8 Type 1m in the Dependent variables edit field.
- 9 Select Lagrange Linear in the Element list.
- IO Click Add.
- II Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- 2 In the **Constants** dialog box, define the following constants with names, expressions, and descriptions (the descriptions are optional). When done, click **OK**.

NAME	EXPRESSION	DESCRIPTION
deltaT	1e-3[K]	Temperature difference
gamma	-8e-5	Temperature derivative of the surface tension
rho1	760[kg/m^3]	Fluid density
eta1	4.94e-4[Pa*s]	Dynamic viscosity
k1	0.1[W/(m*K)]	Thermal conductivity
Cp1	2090[J/(kg*K)]	Heat capacity
alpha1	1.3e-3[1/K]	Thermal expansion coefficient
g	9.8[m/s^2]	Acceleration due to gravity

GEOMETRY MODELING

- I On the Draw menu, point to Specify Objects and then click Rectangle.
- I In the **Rectangle** dialog box, type 0.01 in the **Width** edit field and type 0.005 in the **Height** edit field.
- 2 Click OK.

3 Click the Zoom Extents button on the Main toolbar.

PHYSICS SETTINGS

Boundary Conditions-Momentum Balance

- I From the Multiphysics menu, choose Incompressible Navier-Stokes (ns).
- 2 From the Physics menu, choose Boundary Settings.
- **3** In the **Boundary Settings** dialog box, enter boundary conditions for the Incompressible Navier-Stokes mode according to the table below:

SETTINGS	BOUNDARIES I, 2, 4	BOUNDARY 3
Boundary type	Wall	Symmetry boundary
Boundary condition	No slip	-

4 Click OK.

Point Settings—Momentum Balance

Set the pressure to zero in the point (0,0) to get a well-defined condition on the pressure:

- I From the Physics menu, choose Point Settings.
- 2 In the Point Settings dialog box, select Point 1.
- **3** Select the **Point constraint** check box
- 4 Click OK.

Boundary Conditions-Heat Transfer

- I From the Multiphysics menu, choose Convection and Conduction (cc).
- 2 From the Physics menu, choose Boundary Settings.
- **3** In the **Boundary Settings** dialog box, enter boundary conditions for the Convection and Conduction mode according to the table below:

SETTINGS	BOUNDARY I	BOUNDARIES 2, 3	BOUNDARY 4
Boundary condition	Temperature	Thermal insulation	Temperature
Τ ₀	deltaT		0

4 Click OK.

Boundary Conditions-Liquid/Air Interface

- I From the Multiphysics menu, choose Weak Form, Boundary (wb).
- 2 From the Physics menu, choose Boundary Settings.

- **3** In the **Boundary Settings** dialog box, clear the **Active in this domain** check box for Boundaries 1, 2, and 4.
- 4 Click the **Weak** tab.
- 5 Select Boundary 3 and type lm_test*(eta1*uy-gamma*Tx)+u_test*lm in the weak edit field. This implements Equation 10-8, which states that the shear stress on a surface is proportional to the temperature gradient.
- 6 Click OK.

Boundary Conditions—Equation System Level

To get consistent boundary conditions, omit the Dirichlet boundary condition for T on the vertex connecting Boundary 1 and Boundary 3 and on the vertex connecting Boundary 4 and Boundary 3:

- I On the Physics menu, point to Equation System and then click Boundary Settings.
- 2 In the Boundary Settings Equation System dialog box, click the r tab.
- 3 Select Boundary 1 and type (s<1)*(T0_cc-T) in edit field in row 4. s is the boundary parameterization variable, which varies from 0 to 1 in the direction of the boundary arrow. That means that (s<1) is 1 only at the vertex at the end of the boundary.
- **4** Click the **Differentiate** button to get the *h* coefficient.
- **5** Select Boundary 4 and type (s<1) *T in the edit field in row 4.
- **6** Click the **Differentiate** button to get the *h* coefficient.
- 7 Click OK.

bh system	6
0, g = G, q = $-\partial G/\partial \mathbf{u}$, r = R, h = $-\partial R/\partial \mathbf{u}$	
q g h r Init Element Weak Variables	
r coefficient 0 0 0 (s<1)*T 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
	0, g = G, q = -∂G/∂u, r = R, h = -∂R/∂u q q h r Init Element Weak Variables r coefficient 0 0 -(s<1)*T 0

Subdomain Settings-Momentum Balance

- I From the Multiphysics menu, choose Incompressible Navier-Stokes (ns).
- 2 From the Physics menu, choose Subdomain Settings.
- **3** In the **Subdomain Settings** dialog box, select Subdomain 1 and enter the material properties and buoyancy force term below for the Incompressible Navier-Stokes application mode:

SETTINGS	SUBDOMAIN I
ρ	rho1
η	eta1
Fy	alpha1*g*rho1*(T-deltaT/2)

4 Click OK.

Subdomain Settings-Heat Transfer

- I From the Multiphysics menu, choose Convection and Conduction (cc).
- 2 From the Physics menu, choose Subdomain Settings.
- **3** In the **Subdomain Settings** dialog box, enter the material properties and velocities below for the Convection and Conduction mode:

SETTINGS	SUBDOMAIN I
k (isotropic)	k1
ρ	rho1
C _p	Cp1
u	u
v	v

4 Click OK.

MESH GENERATION

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 In the Free Mesh Parameters dialog box, click the Custom mesh size button and type 1.1 in the Element growth rate edit field.
- **3** Click the **Boundary** tab.
- 4 Select Boundary 3 and type 2e-4 in the Maximum element size edit field.
- 5 Click the **Point** tab.
- 6 Select Point 2 and 4 and type 5e-5 in the Maximum element size edit field.

7 Click **Remesh** and then click **OK**.

These settings provide a finer mesh close to the liquid surface and especially toward the right end.



COMPUTING THE SOLUTION

First solve the problem with the small temperature difference, $\Delta T = 10^{-3}$ K, specified above. Click the **Solve** button to compute the solution. See the section "Postprocessing an Visualization" below on how to plot the temperature and velocity field.

Increase the temperature difference to 0.05 K to make the Marangoni effect more visible:

- I Set deltaT to 0.05[K] in the Constants dialog box.
- 2 Click the **Restart** button.

Finally increase the temperature difference even more (to 2 K):

- I Set deltaT to 2[K] in the **Constants** dialog box.
- 2 Click the **Restart** button

POSTPROCESSING AND VISUALIZATION

Show the temperature field as a surface plot with overlaid contours and the velocity field using arrows as in the results plots earlier in this model documentation:

- I From the **Postprocessing** menu, choose **Plot Parameters**.
- 2 Under Plot type in the Plot Parameters dialog box, select the Surface, Contour, and Arrow check boxes.
- **3** Click the **Surface** tab.
- **4** On the **Surface Data** tab, select **Convection and Conduction (cc)>Temperature** in the predefined quantities list.
- **5** Click the **Contour** tab.
- 6 On the Contour Data tab, select Convection and Conduction (cc)>Temperature in the predefined quantities list.
- **7** In the **Contour color** area, click the **Uniform color** button and then click the **Color** button.
- 8 Select black in the Contour Color dialog box and click OK.
- **9** Click the **Arrow** tab.

IO Verify that the predefined quantity for the arrow data is the velocity field.

II Click OK.

Simulation of a Microrobot²

Introduction

This example describes the modeling of one of the legs of a silicon microrobot in COMSOL Multiphysics. The image below shows the microrobot.

The microrobot uses a technique based on polyimide V-groove joints to get each of the legs to move. The polyimide has a relatively high coefficient of thermal expansion α , and this causes the leg to bend slightly when the polyimide is heated. Putting several V-grooves on each leg provides sufficient deflection (see the following figure).



See Ref. 1 for a more detailed description of the microrobot.

Model Definition

EQUATIONS

The model couples a Heat Transfer application mode to a Plane Strain structural application mode to simulate the bending.

The unknowns of the system are:

- *u*, the displacement along the *x*-axis
- *v*, the displacement along the *y*-axis
- *T*, the temperature

^{2.} Model courtesy of Thorbjörn Ebefors, Dept. of Signals, Sensors and Systems (S3), Royal Institute of Technology (KTH), Stockholm, Sweden.

Time scales for the structural mechanics problem are much smaller than for the heat transfer problem so you can neglect them and analyze the structural model as a static model. The time dependence of the system is retained through the heat transfer equation. This assumption is well motivated and results in a shorter simulation time. The discretized system becomes a differential-algebraic equation system (DAE) containing the following equations:

• The principle of virtual work for plane strain condition states that:

$$\begin{split} \delta W &= \int_{V} (-\varepsilon_{x\text{test}} \sigma_{x} - \varepsilon_{y\text{test}} \sigma_{y} - 2\varepsilon_{xy\text{test}} \tau_{xy} + \mathbf{u}_{\text{test}}^{t} \mathbf{F}_{V}) dV + \\ &\int_{S} \mathbf{u}_{\text{test}}^{t} \mathbf{F}_{S} ds + \int_{S} \mathbf{u}_{\text{test}}^{t} \mathbf{F}_{S} ds = 0 \end{split}$$

where W is the total stored energy. The stress-strain relation including thermal expansion reads:

$$\sigma = D\varepsilon_{el} + \sigma_0 = D(\varepsilon - \alpha T)$$

where α is the thermal expansion coefficient.

See "The Plane Strain Application Mode" on page 217 in the *Modeling Guide* for details.

• The transient heat equation:

$$\rho C_P \frac{\partial T}{\partial t} + \nabla \cdot (k \nabla T) \, = \, Q(t)$$

where ρ is the density, C_P is the heat capacity, k is the thermal conduction, and Q is the heat source.

MATERIAL PROPERTIES

To represent the model use several subdomains corresponding to the different parts. You need six property sets to define each part of the leg of the microrobot (Si, SiNa, Al, P, SiO, and pSi).

THERMAL LOAD

A heat source of $2 \cdot 10^{13}$ W/m³, corresponding to 100 mW, is turned on in each of the two heating resistors during the first 10 ms of the 20 ms simulation.

BOUNDARY CONDITIONS

An important part of the simulation is the modeling of the cooling effect at the boundaries. Experiments have verified that most of the heat dissipates in the silicon structure (the "body" of the robot). Also, the experiments show a minor cooling effect at the tip of the leg.

Motivated by this, use a heat transfer coefficient of $10^6 \text{ W/(m^2 \cdot K)}$ on the part of the leg connected to the rest of the robot, a coefficient of $10^5 \text{ W/(m^2 \cdot K)}$ on the tip of the leg, and insulation on the rest of the structure. These values are somewhat arbitrary, but the simulation results have been calibrated with experimental data.

The boundary conditions for the structural mechanics part are simply constrained displacement at the left end of the robot leg and zero force on the rest of the structure.

Results and Discussion

As expected, the following plots show that the microrobot leg bends when there is a heat source and come back to its initial shape with the decrease of the temperature. The plot sequence shows the temperature increase from 0 s to 20 ms every 5 ms.



The animation shows how the temperature in the robot leg increases and how it bends downward, as expected. Note that the temperature should be interpreted as a deviation from the initial state, which is room temperature. After 10 ms, the power is turned off,

and the leg returns toward its undeformed initial state. It is of great interest to optimize the model, for example, the shape of the V-grooves and the power in each heating resistance, so that the bending process becomes as fast as possible.

Modeling in COMSOL Multiphysics

To model such a system with COMSOL Multiphysics you need to select two application modes: Heat Transfer and Plane Strain.

The coupling appears in the definition of the stress variables by adding the thermal strain.

The coupling is one-way only (from the heat source to the displacements due to the heat source).

Reference

1. T. Ebefors, "Polyimide V-groove Joints for Three-Dimensional Silicon Transducers." Ph.D. thesis, May 2000, Dept. of Signal, Sensors and Systems (S3), Royal Institute of Technology (KTH), Stockholm, Sweden.

Model Library path: COMSOL_Multiphysics/Multiphysics/microrobot

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Start COMSOL Multiphysics or click the New button to open the Model Navigator.
- 2 Select 2D from the Space dimension list.
- 3 Click the Multiphysics button.
- **4** Select the **COMSOL Multiphysics>Heat Transfer>Conduction** application mode and then **Transient analysis**.
- 5 Click Add.
- **6** Open the **Structural Mechanics** application modes folder and then the **Plane Strain** application mode, and select **Static analysis**.
- 7 Click Add.

GEOMETRY MODELING

The geometry of this model is provided as a DXF file as it is quite complicated.

- I On the File menu, point to Import and then click CAD Data From File. Select DXF file (*.dxf) in the Files of type list.
- **2** Browse to models/COMSOL_Multiphysics/Multiphysics under the COMSOL installation directory.
- 3 Select microrobot.dxf. Click Import.
- 4 On the Draw menu, point to Coerce To and then click Solid.



PHYSICS SETTINGS

Because the microrobot legs are composed of several materials with different material properties, you have to define subdomain settings for each materials. Using subdomain expression variables makes it easier to enter the subdomain properties of the model.

Define the following parameters for the materials:

- The thermal conductivity k_mat
- The heat capacity C_mat
- The density: rho_mat

- Young's modulus: E_mat
- The Poisson ratio: nu_mat
- The thermal expansion coefficient: al_mat
- The heat source: heat

Subdomain Expressions

I On the **Options** menu, point to **Expressions** and then click **Subdomain Expressions**.

2 In the Subdomain Expressions dialog box, enter the following material properties:

PROPERTY	SUBDOMAINS 1, 6, 10, 14	SUBDOMAINS 2, 7, 11, 15	SUBDOMAIN 3	SUBDOMAIN 4	SUBDOMAINS 5, 9, 13	SUBDOMAINS 8, 12
k_mat	140	15	238	0.16	1.4	20
C_mat	707	1000	900	2000	1000	707
rho_mat	2300	2865	1410	2700	2200	2300
E_mat	130e9	95e9	70e9	2e9	70e9	160e9
nu_mat	0.3	0.25	0.3	0.35	0.22	0.3
al_mat	2.5e-6	0.8e-6	23e-6	140e-6	0.5e-6	2e-6
heat	0	0	0	0	0	2e13

From left to right the material properties represent Si, SiN, Al, P, SiO, and pSi.

3 Click OK.

Model Settings

The default equation system form for models including the heat transfer application modes is the general form, but this is a linear problem, so you can use the coefficient form. This makes it easier to identify the thermal expansion coefficient in the equation system.

- I From the Physics menu, choose Model Settings.
- 2 Select Coefficient from the Equation system form list.
- 3 Click OK.

Boundary Settings

- I In the Physics menu open the Boundary Settings dialog box.
- 2 Select Boundaries 1, 3, and 5.
- **3** Select the \mathbf{R}_x and \mathbf{R}_y check boxes. This locks the displacement to zero in both the *x* and *y* directions.
- 4 Click OK.

- **5** Change the application mode to **Heat Transfer by Conduction (ht)** in the **Multiphysics** menu.
- 6 Open the Boundary Settings dialog box and enter the following settings:

SETTINGS	ALL BOUNDARIES EXCEPT 1, 3, 5, 95, 96	BOUNDARIES 1, 3, 5	BOUNDARIES 95, 96
Туре	Thermal insulation	Heat flux	Heat flux
h		1e6	1e5

Subdomain settings

- I Change the application mode to Plane Strain (pn) in the Multiphysics menu.
- 2 In the Physics menu, open the Subdomain Settings dialog box.
- 3 Enter following settings for the material properties on the Material page:

SETTINGS	ALL SUBDOMAINS
Ε	E_mat
ν	nu_mat
ρ	0

The density ρ is set to zero to remove all the dynamics in the model.

- 4 Click OK.
- **5** Change the application mode to **Heat Transfer by Conduction (ht)** in the **Multiphysics** menu.
- 6 Open the **Subdomain Settings** dialog box, and set the material properties for the Heat Transfer equation:

SETTINGS	ALL SUBDOMAINS
k (isotropic)	k_mat
ρ	rho_mat
C _p	C_mat
Q	(t<1e-2)*heat

Notice the logical expression (t<1e-2) in the heat source coefficient, which is used to get full heating power (heat) for 10 ms, and after that no power. When done, click **OK**.

Next, include the thermal strain in the stress variables definition:

- 7 On the Physics menu, point to Equations System and then click Subdomain Settings.
- 8 In the Subdomain Settings Equation System dialog box for the equation system, click the Variables tab.
- 9 Select all the subdomains.

IO Change the normal stress variables definition according to the following table:

VARIABLE	DEFINITION
sx_pn	E_pn*(1-nu_pn)*(ex_pn-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))+
	E_pn*nu_pn*(ey_pn-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))+
	E_pn*nu_pn*(-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))
sy_pn	E_pn*nu_pn*(ex_pn-al_mat*T)/(((1+nu_pn)*(1-2*nu_pn))+
	E_pn*(1-nu_pn)*(ey_pn-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))+
	E_pn*nu_pn*(-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))
sz_pn	E_pn*nu_pn*(ex_pn-al_mat*T)/(((1+nu_pn)*(1-2*nu_pn))+
	E_pn*nu_pn*(ey_pn-al_mat*T)/(((1+nu_pn)*(1-2*nu_pn))+
	E_pn*(1-nu_pn)*(-al_mat*T)/((1+nu_pn)*(1-2*nu_pn))

These editings of the stress variables, indicated by the bold font in the previous table, specify the coupling between the heat-transfer problem and the structural mechanics problem.

II Click OK.

MESH GENERATION

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 On the **Global** tab, click the **Custom mesh size** button and type 1.7 in the **Element growth rate** edit field. This reduces the mesh by allowing the element size to grow faster away from a region with small details.
- 3 Click the **Remesh** button to initialize the mesh, and then click **OK**.

COMPUTING THE SOLUTION

- I Open the **Solver Parameters** dialog box by clicking on the corresponding icon in the Main toolbar.
- 2 On the **General** page, type 0:0.005:0.02 in the **Times** edit field. This provides the solution at 5 equally spaced time steps from 0 to 20 ms or every 5 ms. The tolerances for the time-dependent solver control the actual time stepping.
- **3** Type 1e-6 in the **Absolute tolerance** edit field. The absolute tolerance should be smaller than the typical displacements by at least an order of magnitude.

- 4 Click OK.
- 5 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

- I From the Postprocessing menu, open the Plot Parameters dialog box.
- **2** On the **General** page, select the **Deformed shape** check box to plot using a deformed shape.
- **3** Click the **Deform** tab.
- 4 On the Subdomain Data tab in the Deformation data area, select Plane Strain (pn)>Displacement in the Predefined quantities list.
- **5** To get the following plot, clear the **Auto** check box for automatic scaling of the deformed shape. Then type **1** in the **Scale factor** edit field.
- 6 Click OK.
- 7 To get an animation of the results, click the Animate button in the Plot toolbar.



Vibrations in Milk Containers

This example illustrates a multiphysics application that involves a fluid-structure interaction. It simulates a filled milk container on a conveyor belt that starts moving. The methodology presented here defines two subproblems, one for the container walls and another for the fluid. The two problems merge into one using special interface conditions that impose continuous normal displacements and normal forces across the interface boundary.



Model Definition

With all variables having a harmonic time dependence, you solve for only one complex amplitude.

DOMAIN EQUATIONS

In the container walls, use Navier's equation for plane stress to find the displacements *u* and *v*:

$$\begin{cases} -\rho_2 \omega^2 u + \frac{E_2}{1 - v^2} (u_x + vv_y)_x + \frac{E_2}{2(1 + v)} (u_y + v_x)_y = 0\\ -\rho_2 \omega^2 v + \frac{E_2}{1 - v^2} (v_y + vu_x)_y + \frac{E_2}{2(1 + v)} (u_y + v_x)_x = 0 \end{cases}$$

where ρ_2 is density, E_2 is the elasticity modulus (Young's modulus), and v is Poisson's ratio.

For the fluid assume irrotational flow. Thus you can give the displacements with the gradient of a displacement potential G:

$$U = G_x, V = G_y$$

and the equation for sound waves becomes

$$\rho_1 \omega^2 G + E_1 \Delta G = 0$$

where E_1 is the elasticity modulus for the fluid, and ρ_1 the fluid density.

INTERFACE CONDITIONS

At the interface, the normal displacements should be continuous, so

$$\mathbf{n} \cdot \boldsymbol{E}_1 \nabla \boldsymbol{G} - \boldsymbol{E}_1 \mathbf{n} \cdot \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = \mathbf{0}$$

where **n** is the outward-pointing normal. Pressure in the gas, p, follows the relationship

$$p = \rho_1 \omega G$$
,

and force continuity across the interface gives

$$\mathbf{n} \cdot \boldsymbol{\sigma} + \rho_1 \omega^2 G \mathbf{n} = 0$$

where **n** is the inward-pointing normal.

Modeling in COMSOL Multiphysics

This model uses an equation-based approach, where the modeling of both the Navier's equation and the equation for the sound waves uses PDE modes for PDEs in coefficient form. The Navier's equation is active in the wall only, and the equation for

the sound waves is active in the fluid domain only. For the coupling, you use the equation system view to define the fluid-structure interaction at the boundaries between the two domains.

Results

The solution shows the deformations that occur in the container. Figure 10-20 shows the deformations (scaled) for a container filled with milk. A second analysis provides the deformations for a container filled with gas (see Figure 10-21).



Figure 10-20: The deformation (exaggerated) of the container filled with milk.



Figure 10-21: The deformation (exaggerated) of the container filled with gas.

Model Library path: COMSOL_Multiphysics/Multiphysics/milk_container

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, click the Multiphysics button.
- 2 Select the first application mode from the table below in the COMSOL Multiphysics>PDE Modes section and rename it by typing the name in the Application mode name edit field.
- **3** Define the dependent variables by typing the variable names in the table below in the **Dependent variables** edit field.

4 Click the **Add** button.

Model Navigator			X
New Model Library User Models Open Settings			_
Space dimension: 2D	•	Multiphysics	
Heat Transfer	^	Add Remove	
Steady-state analysis Gonduction Structural Mechanics			
PDE Modes Classical PDEs PDE, Coefficient Form	Е		
Cationary analysis Time-dependent analysis Time-dependent analysis Time-dependent analysis, wave type Eigenvalue analysis		Dependent variables: u v Application Mode Properties	
 ⊕ PDE, General Form ⊕ Weak Form, Subdomain ⊕ Weak Form, Boundary 	Ŧ	Add Geometry Add Frame	
Dependent variables: G Application mode name: C Element:		Ruling application mode: PDE, Coefficient Form (wall)	
Lagrange Quander	•	OK Cancel Help	

5 Add the second application mode following the same procedure and then click **OK**.

APPLICATION MODE	APPLICATION MODE NAME	DEPENDENT VARIABLES	ELEMENT TYPE
PDE, coefficient form	wall	u v	Lagrange - Quadratic
PDE, coefficient form	fluid	G	Lagrange - Quadratic

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- 2 In the **Constants** dialog box, add the following constants for the material properties and the angular frequency (the constant w):

NAME	EXPRESSION
E1	1e10
rho1	1e3
E2	1e8
rho2	1e3
nu	0.3
W	2*pi*10

GEOMETRY MODELING

- I Shift-click the **Rectangle/Square** button in the Draw toolbar.
- **2** In the **Rectangle** dialog box, specify a rectangle with a lower-left corner at (0.001, 0.001), a width of 0.063, and a height of 0.163.
- 3 Click OK and then click the Zoom Extents button.
- 4 Draw another rectangle using the same method. This rectangle's lower-left corner is at (0, 0), the width is 0.065, and the height is 0.165.
- 5 Press Ctrl+A to select both objects and then click the Union button on the Draw toolbar.

PHYSICS SETTINGS

Boundary Conditions

- I From the Multiphysics menu, select the wall application mode.
- 2 From the Physics menu, choose Boundary settings.
- 3 In the Boundary Settings dialog box, select the Interior boundaries check box.
- **4** Specify the boundary conditions from the table below. Start with boundary 2. You can the select the **Select by group** check box to set Neumann conditions on the remaining boundaries in one step.

SETTINGS	BOUNDARIES 1,3-8	BOUNDARY 2
Туре	Neumann	Dirichlet
h(I,I)		1
h(1,2)		0
h(2,1)		0
h(2,2)		1
r(1)		0.001
r(2)		0

5 Click OK.

You do not need to define boundary conditions for the fluid because the fluid domain consists of interior boundaries only.

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

- **2** In the **Subdomain Settings** dialog box, select Subdomain 2 from the **Subdomain** selection list and then clear the **Active in this domain** check box to deactivate the fluid domain.
- **3** Select Subdomain 1 and then click the **c**, **a**, and **f** tabs and enter the PDE coefficients for the wall domain according to the following table:

COEFFICIENT	SUBDOMAIN I
c(1,1)	E2/(1-nu^2) E2/2/(1+nu)
c(1,2)	0 E2/2/(1+nu) E2*nu/(1-nu^2) 0
c(2,1)	0 E2*nu/(1-nu^2) E2/2/(1+nu) 0
c(2,2)	E2/2/(1+nu) E2/(1-nu^2)
a(I,I)	-w^2*rho2
a(1,2)	0
a(2,1)	0
a(2,2)	-w^2*rho2
f(1)	0
f(2)	0

- 5 From the Multiphysics menu, select the fluid application mode.
- 6 From the Physics menu, choose Subdomain Settings.
- 7 In the Subdomain Settings dialog box, select Subdomain 1 in the Subdomain selection list and then clear the Active in this domain check box to deactivate the wall domain.
- **8** Select Subdomain 2 and enter the PDE coefficients for the fluid domain according to the following table:

COEFFICIENT	SUBDOMAIN 2
с	E1
a	-w^2*rho1
f	0

Subdomain Settings - PDE, Co	pefficient Form ((fluid)	X			
Equation						
$\nabla \cdot (-c \nabla G \cdot a G + \gamma) + a G + \beta \cdot \nabla G = f$						
Subdomains Groups	Coefficients Ir	nit Element Weak	Color			
Subdomain selection	PDE coefficient	ts				
1	Coefficient	Value/Expression	Description			
2	c	E1	Diffusion coefficient			
	a	-w^2*rho1	Absorption coefficient			
	F	0	Source term			
	ea	0	Mass coefficient			
-	da	1	Damping/Mass coefficient			
Group:	a	0 0	Conservative flux convection coeff.			
	β	0 0	Convection coefficient			
Select by group	Y	0 0	Conservative flux source term			
Active in this domain						
			2			
		ОК	Cancel Apply Help			

Boundary Conditions—Equation System

To get continuous normal forces and normal displacements on the interior boundaries, use the **Boundary Settings** dialog box for the equation system.

- I On the Physics menu, point to Equation System and then click Boundary Settings.
- **2** Specify the following components for the *q* coefficient on Boundaries 4, 5, 6, and 7 that separate the fluid and the wall:

COMPONENT	BOUNDARY 4	BOUNDARY 5	BOUNDARY 6	BOUNDARY 7
q(1,1)	0	0	0	0
q(1,2)	0	0	0	0
q(1,3)	w*rho1	0	0	-w*rho1
q(2,1)	0	0	0	0
q(2,2)	0	0	0	0
q(2,3)	0	w*rho1	-w*rho1	0
q(3,1)	E1	0	0	- E1
q(3,2)	0	E1	-E1	0
q(3,3)	0	0	0	0

3 Click OK.

MESH GENERATION

I From the Mesh menu, choose Free Mesh Parameters.

- 2 Click the **Custom mesh size** button and type 1.7 in the **Element growth rate** edit field.
- 3 Click OK.
- 4 Click the Initialize Mesh button.

COMPUTING THE SOLUTION

For this model, use the stationary solver (the default solver). Click the **Solve** button to solve the problem.

POSTPROCESSING AND VISUALIZATION

- I From the Postprocessing menu, choose Plot Parameters.
- 2 In the Plot type area on the General page, select the Deformed shape check box.
- **3** Click the **Surface** tab.
- 4 On the Surface Data tab, select Geometry and Mesh>y-coordinate from the Predefined quantities list.
- **5** Click the **Deform** tab.
- 6 In the **Deformation data** area, select **PDE**, **Coefficient Form (wall)>u**, **v** from the **Predefined quantities** list on the **Subdomain Data** tab to use the *x* and *y*-displacements for the deformed shape.
- 7 Click OK.

SECOND ANALYSIS-USING A GAS-FILLED CONTAINER

To model a container filled with gas instead of milk, update the fluid parameters:

OPTIONS AND SETTINGS

Change the following constants in the **Constants** dialog box:

NAME EXPRESSIO	
E1	1e5
rho1	1

COMPUTING THE SOLUTION

Solve the problem again with the new material properties for the fluid. You get a similar plot of the solution from the second analysis.

Peristaltic Pump

Introduction

In a peristaltic pump, rotating rollers are squeezing a flexible tube. As the pushed down rollers move along the tube, the fluid in the tube follows the motion. The main advantage of the peristaltic pump is that no seals, valves, or other internal parts ever touch the fluid. Due to their cleanliness, peristaltic pumps have found many applications in the pharmaceutical, chemical, and food industries. Besides this, the action of a peristaltic pump is very gentle, which is important if the fluid is easily damaged. Peristaltic pumps are therefore used in medical applications, one of which is moving the blood through the body during open heart surgery. Other pumps would risk destroying the blood cells.

This COMSOL Multiphysics model of a peristaltic pump is a combination of structural mechanics, for modeling the squeezing of the tube, and fluid dynamics, for computing the motion of the fluid (fluid-structure interaction, FSI). As it is the deformations of the tube that drive the fluid, the fluid part of the simulation needs to be done in a deforming geometry, defined by COMSOL Multiphysics' Moving Mesh (ALE) application mode.

Model Definition

The model is set up in 2D axial symmetry (Figure 10-22). A vinyl tube of length 0.1 m, inner radius 1 cm, and outer radius 1.5 cm, contains a gas of density $\rho = 1 \text{ kg/m}^3$ and viscosity $\eta = 3 \cdot 10^{-5}$ Pa·s. A time- and position-dependent force density is applied to the outer wall of the tube, in the radial direction. This force density could have been taken from real data from a peristaltic pump, but is for simplicity modeled as a Gaussian distribution along the length of the tube. The Gaussian has a width of 1 cm and is moving with the constant velocity 0.03 m/s in the positive *z* direction. To represent the engagement of the roll, the force density, multiplied by a smoothed Heaviside function, kicks in at t = 0.1 s and takes the force to its full development at t = 0.5 s.

Likewise, the disengagement of the roll starts at t = 1.0 s and ends at t = 1.4 s. The deformations of the tube are modeled during a full cycle of 1.5 s.



Figure 10-22: The geometry of the peristaltic pump in its initial shape (left) and as it is deforming under the pressure of the roll. The tube is rotationally symmetric with respect to the z axis. The color scale shows the deformation of the tube material.

DOMAIN EQUATIONS

The structural mechanics computations are done using a 2D axially symmetric stress-strain formulation. The model uses the assumption that the material is linear and elastic and does not take geometric nonlinearities into account. With the capabilities in the COMSOL Structural Mechanics Module, you could easily extend this model to include large deformations and a more complicated material model.

The fluid flow is described by the Navier-Stokes equations:

$$\rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0$$
(10-9)
$$\nabla \cdot \mathbf{u} = 0$$

where ρ denotes the density (kg/m³)(, **u** the velocity vector (m/s), η the viscosity (Pa·s), and *p* the pressure (Pa). The equations are set up and solved in axial symmetry on a deformed mesh inside the tube.

The motion of the deformed mesh is modeled using Winslow smoothing. For more information, please refer to "The Moving Mesh Application Mode" on page 401 in the *COMSOL Multiphysics Modeling Guide*.

BOUNDARY CONDITIONS

For the structural mechanics computations, the time and coordinate dependent load described above is the boundary condition at the outer surface of the tube. Whereas the top and bottom ends of the tube are constrained along both coordinate axes, the inner surface is free. The reaction forces from the fluid on the tube are very small, so this model does not take them into account. This way you can solve first for the structural mechanics and then for the fluid dynamics, reducing solution time and memory consumption.

The inner surface is free, and the top and bottom ends of the tube are constrained along both coordinate axes.

For the fluid simulation, the boundary condition at the inlet and the outlet assumes that the total stress is zero, that is:

$$\mathbf{n} \cdot [-pI + \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] = \mathbf{0}$$
(10-10)

Inside the wall of the tube, the moving mesh follows the deformations of the tube. The mesh deformations being the same as the structural deformations is also used as the moving mesh boundary condition where the fluid meets the tube wall. The mesh is fixed to a zero r displacement at the symmetry axis and a zero z displacement at the top and the bottom of the tube.

Results

Figure 10-23 shows three snapshots from the peristaltic pump in action, and Figure 10-25 shows the inner volume of the tube as a function of time. At t = 0.3 s, the roll has begun its engagement phase and is increasing its pressure on the tube. As less space is left for the gas, it is streaming out of the tube, through both the inlet and the outlet. At t = 0.7 s, the roll has been fully engaged for a while. As it is moving up the tube, so is the gas, both at the inlet and at the outlet. This is where most of the net flow in the direction from the inlet to the outlet is created. Finally, at t = 1.2 s, the engagement process is reversed as the roll is disengaging. Fluid is streaming into the tube from both ends.



Figure 10-23: Snapshots of the velocity field and the shape of the inside of the tube at t = 0.3 s, t = 0.7 s, and t = 1.2 s. The colors represent the magnitude of the velocity, and the arrows its direction.



Figure 10-24: The inner volume (m3) of the tube as a function of time (s).

Figure 10-25 shows the inlet and outlet flows. The overall behavior indicated in the velocity snapshots is confirmed. As the roll is engaged, gas is squeezed out through both the inlet and the outlet. When the fully engaged roll is moving along the tube, gas is steadily flowing in through the inlet and out through the outlet. As the roll is disengaging, the gas is being sucked back into the tube. In a real peristaltic pump, the peaks associated with the volume changes are usually removed or kept to a minimum,

by another roll engaging at the same time as the first roll is disengaging. This way, there are hardly any volume changes, and the fluid is flowing forward all the time.



Figure 10-25: Inlet (blue solid line) and outlet (red dashed line) flow in m^3 /s as functions of time. Positive values indicate that the gas is flowing in through the inlet and out through the outlet.

Figure 10-26 sums up the process, plotting the accumulated net flow versus time. It is worth noting that although the accumulated flow during the first 0.5 s of the cycle is zero or negative, it is well above zero after the full cycle.



Figure 10-26: Accumulated flow (m^3) through the pump versus time (s).

Modeling in COMSOL Multiphysics

This model is primarily intended to demonstrate the use of the Moving Mesh (ALE) application mode, but it also shows some postprocessing features. Integration coupling variables are defined for the flow computations and for evaluating the inside volume of the tube. An ordinary differential equation is used for calculating the accumulated flow. The smooth step function in this model is flc2hs, which is a C²-continuous step with a given step width.

Model Library path: COMSOL_Multiphysics/Multiphysics/peristaltic_pump

Modeling Using the Graphical User Interface

I Start COMSOL Multiphysics.

- 2 In the Model Navigator, set the Space dimension to Axial symmetry (2D) and click the Multiphysics button.
- 3 Select COMSOL Multiphysics>Structural Mechanics>Axial Symmetry, Stress-Strain>Transient analysis in the list of application modes and click Add.
- 4 Select COMSOL Multiphysics>Deformed Mesh>Moving Mesh (ALE)>Transient analysis and click Add.
- 5 Click the Application Mode Properties button and set the Smoothing method to Winslow. Click OK.
- 6 Select COMSOL Multiphysics>Fluid Dynamics>Incompressible Navier-Stokes>Transient analysis and click Add.
- 7 Click OK to close the Model Navigator.

GEOMETRY MODELING

8 Specify two rectangles according to the table below

NAME	WIDTH	HEIGHT	BASE	R	z
RI	0.01	0.1	Corner	0	0
R2	5e-3	0.1	Corner	0.01	0

9 Click the **Zoom Extents** button in the Main toolbar.

OPTIONS AND SETTINGS

I Open the Constants dialog box from the Options menu and enter constants according to the following table. The descriptions are optional. When done, click OK.

NAME	EXPRESSION	DESCRIPTION
t_on	0.3[s]	Time when roll is engaged
t_off	1.2[s]	Time when roll is disengaged
dt	0.2[s]	Time to reach full force
z0	0.03[m]	z coordinate where roll starts
v0	0.03[m/s]	Vertical velocity of roll
width	0.01[m]	Width of Gaussian force distribution
Ttot	1.5[s]	Total time for a pump cycle
Lmax	4e6[N/m^2]	Max load

2 In the **Options** menu, go to **Functions** and define a new function according to following table. When done, click **OK**.

NAME	ARGUMENTS	EXPRESSION
gauss	x, sigma	1/(sigma*sqrt(2*pi))*exp(-x^2/(2*sigma^2))

3 Go to **Physics>Space-Independent Equations** and define an ordinary differential equation with the following settings, When done, click **OK**.

NAME (U)	EQUATION	INIT(U)	INIT (UT)
netflow	<pre>netflowt-(inflow+outflow)/(2*Ttot)</pre>	0	0

4 Select **Integration Coupling Variables>Boundary Variables** from the **Options** menu and specify the following boundary integration variables, each on a separate row in the table. When done, click **OK**.

SOURCE BOUNDARY	NAME	EXPRESSION	INTEGRATION ORDER	GLOBAL DESTINATION
2	inflow	2*pi*r*v	4	checked
3	outflow	2*pi*r*v	4	checked

5 Select **Integration Coupling Variables>Subdomain Variables** from the **Options** menu and specify the following subdomain integration variables. When done, click **OK**,.

SOURCE SUBDOMAIN	NAME	EXPRESSION	INTEGRATION ORDER	GLOBAL DESTINATION
1	Vol	2*pi*r	4	checked

PHYSICS SETTINGS—STRESS-STRAIN

Subdomain Settings

- I In the Multiphysics menu, select Axial Symmetry, Stress-Strain.
- **2** Open the **Subdomain Settings** dialog box. Select Subdomain 1 and clear the **Active in this domain** check box.
- 3 Select Subdomain 2 and click Load. Select Nylon from the list of materials and click OK.
- 4 Click OK to close the Subdomain Settings dialog box.

Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu and apply the following boundary settings. When done, click **OK**.

SETTINGS	BOUNDARY 4	BOUNDARIES 5, 6	BOUNDARY 7
$Constraint\ R_R$		0	
$Constraint\ R_{Z}$		0	
Load F _R			See below
Load F _Z			0

For the F_r load on Boundary 7, enter the following expression:

-Lmax*flc2hs((t-t_on)[1/s],dt[1/s])*gauss((z-z0-v0*t)[1/m], width[1/m])*flc2hs((t_off-t)[1/s],dt[1/s])

The reason for the unit conversions [1/s] and [1/m] is to make the input to the flc2hs and gauss functions dimensionless.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box from the Mesh menu.
- **2** On the **Global** page, click the **Custom mesh size** button and set **Maximum element size** to 2e-3.
- 3 Click Remesh and then OK.

COMPUTING THE SOLUTION

- I In the **Solver Manager**, go to the **Solve For** tab and select **Axial Symmetry, Stress-Strain**. Only this application mode and its solution variables **uor** and **w** should be selected. Click **OK** to close the dialog box.
- 2 In the Solver Parameters dialog box, select Time dependent from the Solver list. Set Times to 0:0.01:1.5, Relative tolerance to 0.001 and Absolute tolerance to 1e-5.
- **3** On the Time Stepping tab, set Time steps taken by solver to Strict.
- 4 Click **OK** and then **Solve**.

POSTPROCESSING AND VISUALIZATION

The default plot shows the von Mises stress distribution.

- I In the Plot Parameters dialog box, click the Deform tab.
- 2 Select the **Deformed shape plot** check box and set the **Scale factor** to 1.

3 Use the **Solution at time** list on the **General** tab and click **Apply** to study the stress distribution and the deformation of the tube at a few points in time.

PHYSICS SETTINGS—INCOMPRESSIBLE NAVIER-STOKES

Subdomain Settings

- I In the Multiphysics menu, select Incompressible Navier-Stokes.
- 2 Open the Subdomain Settings dialog box. Select Subdomain 2 and clear the Active in this domain check box.
- **3** In Subdomain 1, set the **Density** to 1 (kg/m^3) and the **Dynamic viscosity** to 3e-5 (Pa·s).
- 4 Click OK to close the Subdomain Settings dialog box.

Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu and apply the following boundary settings.

SETTINGS	BOUNDARY I	BOUNDARY 2	BOUNDARY 3	BOUNDARY4
Туре	Symmetry	Open	Open	Inlet
Condition	Axial symmetry	Normal stress	Normal stress	Velocity
u ₀				uaxi_t_axi
v ₀				wt
f ₀		0	0	

PHYSICS SETTINGS-MOVING MESH

Subdomain Settings

- I In the Multiphysics menu, select Moving Mesh (ALE).
- **2** Open the **Subdomain Settings** dialog box. Keep the default **Free displacement** in Subdomain 1.
- 3 In Subdomain 2, select **Prescribed displacement**. Enter uaxi_axi for dr and w for dz. These are the structural r and z deformations. The reason you cannot use physics-induced displacements here is that uaxi_axi is not a solution variable in the Stress-Strain application mode.
- 4 Click OK to close the Subdomain Settings dialog box.
Boundary Conditions

Open the **Boundary Settings** dialog box from the **Physics** menu and apply the following boundary settings.

SETTINGS	BOUNDARY I	BOUNDARIES 2–3	BOUNDARY 4
Туре	Mesh displacement	Mesh displacement	Mesh displacement
dr	0		uaxi_axi
dz		0	W

COMPUTING THE SOLUTION

- I In the Solver Manager, go to the Initial Value tab and click the Store Solution button. By default, all solutions are selected in the Store Solution dialog box. Click OK to confirm the selection.
- 2 In the Values of variables not solved for and linearization point area, click the Stored solution option button and select All in the Solution at time list. This makes the solver use the solution for the deformation of the tube when computing the fluid flow.
- 3 On the Solve For tab, select all variables but uor and w. Click OK.
- 4 Open the Solver Parameters dialog box, go to the General tab. Set the Absolute tolerance to 1e-3 and the Relative tolerance to 0.01.
- 5 On the Time Stepping tab, set the Error estimation strategy to Exclude algebraic.
- 6 Click **OK** and then **Solve**.

POSTPROCESSING AND VISUALIZATION

The plot still shows the deformations from the stress-strain computation. To see the velocity field as in Figure 10-23 on page 411, do the following:

- I In the **Plot Parameters** dialog box, go to the **General** tab and clear the **Deformed shape** check box.
- 2 On the Surface tab, select Incompressible Navier Stokes (ns)>Velocity field among the Predefined quantities.
- **3** On the **Arrow** tab, select **Incompressible Navier Stokes (ns)>Velocity field** among the **Predefined quantities**. Click the **Color** button and choose a black color; then click **OK**. Select the **Arrow plot** check box and click **Apply**.
- **4** Once more, you can select different times to look at on the **General** tab. You will see the deformations even though you are not making a deformation plot, because the mesh is deformed.

- **5** If you want to animate the velocity field as a function of time, go to the **Animate** tab and click **Start Animation**. It may take some time for your computer to render the animation.
- 6 After closing the Plot Parameters dialog box, go to Postprocessing> Domain Plot Parameters.
- 7 Use the **Expression** field on the **Point** tab to plot, for instance, the expressions inflow, outflow, netflow, or Vol. Because these variables are defined with a global destination and do not depend on the location, the coordinates do not matter. If you want to show more than one curve at the same time, use the **Keep current plot** check box on the **General** page.

Quantum Mechanics Models

 $T_{\rm his}$ chapter contains quantum mechanics models that center on solutions of the Schrödinger equation.

The Schrödinger Equation for the Hydrogen Atom

Introduction

This example shows how to compute energy levels and electron orbits for the hydrogen atom. It models the atom as a 1-particle system using the stationary Schrödinger equation.

$$-\nabla \cdot \left(\frac{h^2}{8\mu\pi^2}\nabla\Psi\right) + V\Psi = E\Psi$$

The equation parameters are:

- $h (6.626 \cdot 10^{-34} \text{ Js})$ is Planck's constant.
- μ is the reduced mass
- V is the potential energy.
- *E* is the unknown energy eigenvalue.
- Ψ is the quantum mechanical wave function.

Model Definition

The quantity $|\Psi|^2$ corresponds to the probability density function of the electron's position. In this example,

$$\mu = \frac{Mm_e}{M+m_e} \approx m_e$$

where *M* equals the mass of the nucleus and m_e represents the mass of an electron $(9.1094 \cdot 10^{-31} \text{ kg})$. The hydrogen nucleus consists of a single proton (more than 1,800 times heavier than the electron), so the approximation of μ is valid. Thus you can treat the problem as a one-particle system.

The system's potential energy is

$$V = -\frac{e^2}{4\pi\varepsilon_0}$$

where *e* equals the electron charge (1.602e-19 C), ε_0 represents electric capacitivity in a vacuum (8.854e-12 F/m), and *r* gives the distance from the center of the atom.

Results

The solution provides a number of the lowest eigenvalues.



Three quantum numbers (n, l, m) characterize the eigenstates of a hydrogen atom:

- *n* is the principal quantum number.
- *l* is the angular quantum number.
- *m* is the magnetic quantum number that appears in the equations above.

These quantum numbers are not independent but have the following mutual relationships:

$$n = 1, 2, 3, ...$$

 $0 \le l \le n - 1$
 $-l \le m \le l$.

An analytical expression exists for the energy eigenvalues in terms of the quantum number n

$$E_n = -\frac{h^2}{8\pi^2 \mu a_0^2 n^2}$$

where

$$a_0 = \frac{h^2 \varepsilon_0}{\pi \mu e^2}.$$
 (11-1)

This expression is called the Bohr radius and has an approximate value of $3 \cdot 10^{-11}$ m.

The first three energy eigenvalues, according to the above expression with $\mu \approx m_e$, are:

- $E_1 \approx -2.180 \cdot 10^{-18} \text{ J}$
- $E_2 \approx -5.450 \cdot 10^{-19} \text{ J}$
- $E_3 \approx -2.422 \cdot 10^{-19} \text{ J}$

Comparing these numbers with the computed eigenvalues, you can see a 2-fold degeneracy for n = 2 and a 3-fold degeneracy for n = 3. This degeneracy corresponds to the following quantum triplets (2,0,0), (2,1,0) and (3,0,0), (3,1,0), (3,2,0). The computed values are separated due to the approximate numerical solution.

By refining the mesh and solving again, the results are more accurate. The states with l = 0 correspond to spherically symmetric solutions, while states with l = 1 or 2 correspond to states with one or two radial node surfaces. The 0 energy level corresponds to the energy of a free electron no longer bounded to the nucleus. Energy levels closer to 0 correspond to excited states.

The wave function by itself has no direct physical interpretation. Another quantity to plot is $|u|^2$, which is proportional to the probability density (unnormalized) function $|\Psi|^2$ for the electron position after integration about the *z*-axis. The plot shows the unnormalized probability density function.

You can now solve the problem for a different integer value of m as well as with a different eigenvalue search range. Remember that high energy levels correspond to large radial components in the solution, so you may have to enlarge the geometry to produce correct results. Also, try to solve the problem using a asymmetric domain.

To determine the ground state energy, use the adaptive solver. Using as many elements as before, the software now adapts the mesh to the first eigenfunction in an attempt to equidistribute the error. This yields an approximation of the ground state energy with

a relative error less than 1e-5, about two orders of magnitude better than the error in the solution using the initial mesh.



Modeling in COMSOL Multiphysics

To solve this problem in COMSOL Multiphysics, reduce the dimension of the problem from three to two by using cylindrical coordinates (ρ , ϕ , z). The Schrödinger equation transforms into

$$-\frac{\hbar^2}{8\mu\pi^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Psi}{\partial\rho}\right) + \frac{1}{\rho^2}\frac{\partial^2\Psi}{\partial\phi^2} + \frac{\partial^2\Psi}{\partial z^2}\right) + V\Psi = E\Psi$$

and you can write the potential energy as

$$V(\rho, z) = -\frac{e^2}{4\pi\varepsilon_0\sqrt{(\rho^2 + z^2)}}$$

Consider the wave function Ψ as the product of two functions, one with an angular dependence and the other with a radial and *z*-coordinate dependence,

$$\Psi = \Psi_1(\varphi)\Psi_2(\rho, z)$$

Substituting this expression into the equation gives

$$-\frac{\hbar^2}{8\mu\pi^2}\left(\Psi_1\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Psi}{\partial\rho}^2\right) + \Psi_2\frac{1}{\rho^2}\frac{\partial^2\Psi_1}{\partial\phi^2} + \Psi_1\frac{\partial^2\Psi_2}{\partial z^2}\right) + V\Psi_1\Psi_2 = E\Psi_1\Psi_2$$

Multiplying by $(8\mu\pi^2/\hbar^2)
ho^2/(\Psi_1\Psi_2)$ and rearranging terms results in

$$-\left(\frac{\rho}{\Psi_2}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Psi_2}{\partial\rho}\right) + \frac{\rho}{\Psi_2}\frac{\partial}{\partial z}\left(\rho\frac{\partial\Psi}{\partial z}^2\right)\right) + C\rho^2 V - C\rho^2 E = \frac{1}{\Psi_1}\frac{\partial^2\Psi_1}{\partial\varphi^2},$$

where $C = 8\mu\pi^2/h^2$, or equivalently:

$$-\left(\frac{\rho}{\Psi_2}(\nabla \cdot (\rho \nabla \Psi_2))\right) + C\rho^2 V(\rho, z) - C\rho^2 E = \frac{1}{\Psi_1} \frac{\partial^2 \Psi_1}{\partial \varphi^2}$$

The left side is independent of φ , and the right side is independent of z and ρ , so the equation is equal to a constant, which you can call $-m^2$. Thus you can split the equation into two parts:

$$-\left(\frac{\rho}{\Psi_2}(\nabla \cdot (\rho \nabla \Psi_2))\right) + C\rho^2 V(\rho, z) - C\rho^2 E = -m^2$$

and

$$\frac{1}{\Psi_1} \frac{\partial^2 \Psi_1}{\partial \varphi^2} = -m^2$$

The equation for Ψ_1 gives the solution

$$\Psi_1 = A\sin(m\varphi) + B\cos(m\varphi),$$

where *A* and *B* are arbitrary real constants, and $m = 0, \pm 1, \pm 2, ...$ due to the periodic boundary condition $\Psi_1(0) = \Psi_2(2\pi k)$, k = any integer.

Using the equation for Ψ_2 with *m* included and multiplying by Ψ_2/ρ , you arrive at

$$-\left(\nabla\cdot(\rho\nabla\Psi_{2})\right)+\left(C\rho V(\rho,z)+\frac{m^{2}}{\rho}\right)\Psi_{2}=C\rho E\Psi_{2}$$

The equation now exists in a form that you can immediately transfer into COMSOL Multiphysics. The coordinate variables *x* and *y* replace ρ and *z*, and Ψ_2 becomes *u*.

COMSOL Multiphysics describes the coefficient form eigenvalue PDE as

$$-\nabla \cdot (c\nabla u + \alpha u) + \beta \nabla u + au = d_a \lambda u \quad .$$

For this problem, use the following coefficient values:

$$\left\{ \begin{array}{l} c = x \\ a = -C \frac{xe^2}{4\pi\varepsilon_0 \sqrt{x^2 + y^2}} + \frac{m^2}{x} \\ d_a = Cx \end{array} \right. \label{eq:alpha}$$

Set all other coefficient values to 0.

The geometry consists of two concentric semicircular solids. The circles have centers at (0, 0) with radii of $3 \cdot 10^{-9}$ m and $5 \cdot 10^{-10}$ m. Remove the portions to the left of the *y*-axis by subtracting a rectangular solid with corners at $(-1 \cdot 10^{-9}, -1 \cdot 10^{-9})$ and $(0, 1 \cdot 10^{-9})$, to avoid negative *x*-values, because the use of cylindrical coordinates makes ρ positive.

The boundary conditions are as follows:

- The cylinder axis x = 0 is not a boundary in the original problem, but here it becomes one. You must give the artificial boundary condition n · (c\overline{vu}) = 0. Because c = 0 on the cylinder axis, you should not view this boundary condition as a Neumann condition that imposes a constraint. Rather it indicates the absence of constraints.
- On the outer circular boundary, the Dirichlet condition u = 0 applies.

The Dirichlet condition corresponds to a zero probability for the electron to be outside the specified domain. This means that the probability of finding the electron inside the domain is 1. It is important to have this approximation in mind when solving for higher-energy eigenvalues because the solution of the physical problem might fall outside the domain, and no eigenvalues are found for the discretized problem. Ideally the domain is infinite and higher-energy eigenvalues correspond to the electron being further away from the nucleus.

Model Library path:

COMSOL_Multiphysics/Quantum_Mechanics/hydrogen_atom

MODEL NAVIGATOR

- I Select 2D from the Space dimension list.
- 2 In the list of application modes, open COMSOL Multiphysics>PDE Modes and then PDE, Coefficient Form.
- **3** Select **Eigenvalue analysis**. Make sure that **Lagrange Quadratic** elements are selected in the **Element** list.
- 4 Click OK.

OPTIONS AND SETTINGS

- I Open the Axes/Grid Settings dialog box.
- **2** Set axis and grid values:

AXIS		GRID	
x min	-4.5e-9	x spacing	1e-9
x max	4.5e-9	Extra x	
y min	-3.1e-9	y spacing	1e-9
y max	3.1e-9	Extra y	5e-10

3 Open the **Constants** dialog box.

4 Enter the following constants and expressions:

NAME	EXPRESSION
me	9.10939e-31
е	1.6021773e-19
hp	6.626076e-34
e0	8.8541878e-12
С	(8*me*pi^2)/hp^2
m	0

These names correspond to the constants $m_e, e, h, \varepsilon_0, C$, and the quantum number m. First set m equal to 0 to catch the ground state energy among the energy eigenvalues.

GEOMETRY MODELING

I Draw a circle centered at (0, 0) with a radius of 3 x 10⁻⁹.

- 2 Draw a circle centered at (0, 0) with a radius of $0.5 \cdot 10^{-9}$. Use the tick mark at $y = 0.5 \cdot 10^{-9}$ to assist in placing the radius correctly.
- **3** Draw a rectangle from $(-3 \cdot 10^{-9}, -3 \cdot 10^{-9})$ to $(0, 3 \cdot 10^{-9})$.
- 4 From the Draw menu, choose Create Composite Object.
- 5 Enter (C1+C2) -R1 in the Set formula edit field.
- 6 Click OK.

PHYSICS SETTINGS

Boundary Conditions

I Open the Boundary Settings dialog box.

2 Enter boundary coefficients as indicated in the table below:

SETTINGS	BOUNDARIES 1-3	BOUNDARIES 4,7
Туре	Neumann	Dirichlet
q	0	0
g	0	0
h		1
r		0

Subdomain Settings

I Open the Subdomain Settings dialog box.

2 Enter the following PDE coefficients.

COEFFICIENT	ALL SUBDOMAINS
с	x
a	-C*(x*e^2)/(4*pi*e0*sqrt(x^2+y^2))+(m^2)/x
d _a	C*x

MESH GENERATION

Use a finer mesh in the inner circular domain, because the solution shows greater variations in the center region than in the outer regions for low-energy eigenvalues.

- I Open the Free Mesh Parameters dialog box
- 2 Click the **Custom mesh size** button and type 1.1 in the **Element growth rate** edit field.
- **3** Click the **Subdomain** tab.

- 4 Select Subdomain 2 and enter 0.05e-9 in the Maximum element size edit field. This ensures that Subdomain 2 (the inner domain) has a finer mesh than the outer subdomain.
- **5** Click the **Remesh** button.
- 6 Click OK.

COMPUTING THE SOLUTION

- I Open the Solver Parameters dialog box.
- 2 On the General page, type 10 in the Desired number of eigenvalues edit field.
- 3 Type -2e-18 in the Search for eigenvalues around edit field.
- 4 Click OK.
- 5 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The default plot shows the lowest energy eigenvalue, or ground state, of approximately $-2 \cdot 10^{-18}$ J.

- I Open the Plot Parameters dialog box.
- 2 Plot some of the eigenfunctions by selecting the corresponding eigenvalue from the **Eigenvalue** list on the **General** page.
- **3** Click the **Surface** tab.
- 4 On the Surface Data tab, type abs(u)² in the Expression edit field.
- 5 Click **OK** to plot the probability density function.

Finally, try to determine the ground state energy accurately by enabling the adaptive solver. Use approximately as many elements as before, but the solver now adapts the mesh for the first eigenvalue in an attempt to equidistribute the error.

MESH GENERATION

- I Open the Free Mesh Parameters dialog box.
- 2 On the General page, click the Custom mesh size button and type 1.5 in the Element growth rate edit field.
- **3** Click the **Subdomain** tab and clear the entry in the **Maximum element size** edit field for Subdomain 2.
- 4 Click the **Remesh** button and then click **OK**.

COMPUTING THE SOLUTION

Search for a single eigenvalue using the approximate ground state energy -2e-18 as shift. This ensures that the eigenvalue solver finds the lowest state.

- I Open the Solver Parameters dialog box.
- 2 On the General page, select the Adaptive mesh refinement check box.
- **3** Type **1** in the **Desired number of eigenvalues** edit field.
- **4** Click the **Adaptive** tab.
- 5 Type 2500 in the Maximum number of elements edit field.
- 6 Type 10 in the Maximum number of refinements edit field.
- 7 Click OK.
- 8 Click the Solve button on the Main toolbar.

Conical Quantum Dots

This model computes the electronic states for a quantum-dot/wetting-layer system. It was inspired largely from the work of Dr. M. Willatzen and Dr. R. Melnik (Ref. 1) as well as B. Lassen.

Introduction

Quantum dots are nano- or microscale devices created by confining free electrons in a 3D semiconducting matrix. Those tiny islands or droplets of confined "free electrons" (those with no potential energy) present many interesting electronic properties. They are of potential importance for applications in quantum computing, biological labeling, and lasers, to name only a few.

Scientists can create such structures experimentally using the Stranski-Krastanow molecular beam-epitaxy technique. In that way they obtain 3D confinement regions (the quantum dots) by growth of a thin layer of material (the wetting layer) onto a semiconducting matrix. Quantum dots can have many geometries including cylindrical, conical, or pyramidal. This model studies the electronic states of a conical InAs quantum dot grown on a GaAs substrate.

To compute the electronic states taken on by the quantum dot/wetting layer assembly embedded in the GaAs surrounding matrix, you must solve the 1-band Schrödinger equation in the effective mass approximation:

$$-\frac{h^2}{8\pi^2} \left(\nabla \cdot \left(\frac{1}{m_e(r)} \nabla \Psi(r) \right) \right) + V(r) \Psi(r) = E \Psi(r)$$

where *h* is Planck's constant, Ψ is the eigenwave function, *E* is the eigenvalue (energy), and m_e is the effective electron mass (to account for screening effects).

Model Definition

The model works with the 1-particle stationary Schrödinger equation

$$-\nabla \cdot \left(\frac{h^2}{8m\pi^2}\nabla\Psi\right) + V\Psi = E\Psi$$

It solves this eigenvalue problem for the quantum-dot/wetting-layer system using the step potential barrier and effective-mass approximations: V = 0 for the InAs quantum dot/wetting layer and V = 0.697 eV for GaAs.

The effective masses m_e are 0.023m for InAs and 0.067m for GaAs.

Assume the quantum dot has perfect cylindrical symmetry. In that case you can model the overall structure in 2D as shown in the following figure.



2D geometry of a perfectly cylindrical quantum dot and wetting layer.

You can now separate the total wave function Ψ into

$$\Psi = \chi(z, r)\Theta(\varphi)$$

where ϕ is the azimuthal angle. Then rewrite the Schrödinger equation in cylindrical coordinates as

$$-\frac{h^2}{8\pi^2} \left[\frac{\partial}{\partial z} \left(\frac{1}{m_e} \frac{\partial \chi}{\partial z}\right) - \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r}{m_e} \frac{\partial \chi}{\partial r}\right)\right] \Theta - \frac{h^2}{8m_e \pi^2} \frac{\chi}{r} \frac{\partial^2 \Theta}{\partial \varphi^2} + V_e \chi \Theta = E \chi \Theta$$

Dividing this equation by

$$\frac{\chi(z,r)}{m_e r^2} \Theta(\varphi)$$

and rearranging its terms lead to the two independent equations

$$-m_e r^2 \frac{h^2}{8\pi^2} \left[\frac{\partial}{\partial z} \left(\frac{1}{m_e} \frac{\partial \chi_l}{\partial z} \right) \frac{1}{\chi_l} + \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r}{m_e} \frac{\partial \chi_l}{\partial r} \right) \frac{1}{\chi_l} \right] + m_e r^2 [V_e - E] = -\frac{h^2}{8\pi} l^2 \quad (11-2)$$

and

$$\frac{h^2}{8\pi^2}\frac{\partial^2\Theta}{\partial\varphi^2}\frac{1}{\Theta} = -\frac{h^2}{8\pi^2}l^2$$
(11-3)

Equation 11-3 has obvious solutions of the form

 $\Theta = \exp[il\varphi]$

where φ acts as a phase factor, and *l* is the principal quantum number. You also need to solve Equation , which for clarity you can rewrite as

$$-\frac{h^2}{8\pi^2}\frac{\partial}{\partial z}\left(\frac{1}{m_e}\frac{\partial\chi_l}{\partial z}\right) - \frac{h^2}{8\pi^2}\frac{1}{r}\frac{\partial}{\partial r}\left(\frac{r}{m_e}\frac{\partial\chi_l}{\partial r}\right) + \frac{h^2}{8m_e\pi^2}\frac{l^2}{r^2}\chi_l + V_e\chi_l = E_l\chi_l$$

Note that this is an instance of a PDE on coefficient form,

$$\nabla \cdot \left(\left(-c \nabla u - \alpha u + \gamma \right) + a u + \beta \cdot \nabla u \right) = d_a \lambda u$$

where the coefficients become

$$\label{eq:constraint} \begin{split} c &\to \frac{h^2}{8\pi^2 m_e} \\ a &\to \frac{h^2}{8\pi^2 m_e x^2} l^2 + V_e \end{split}$$

and

$$\beta_x \rightarrow -\frac{h^2}{8\pi^2 m_e x}$$

Results

This exercise models the eigenvalues for the four lowest electronic energy levels corresponding to a value of l = 0. The following plots show the eigenwave functions for those four states:



The four lowest electronic-energy levels corresponding to a value of l = 0.

Modeling in COMSOL Multiphysics

To solve this problem use the PDE, Coefficient Form version of the PDE modes in the Model Navigator. The model solves for an eigenvalue/eigenfunction, for which you must input appropriate physical data and constants.

Reference

1. R. Melnik and M. Willatzen, "Band structure of conical quantum dots with wetting layers," *Nanotechnology*, vol. 15, pp. 1–8, 2004.

Model Library path:

COMSOL_Multiphysics/Quantum_Mechanics/conical_quantum_dot

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Open the Model Navigator. In the Space dimension list select 2D.
- 2 In the list of application modes select COMSOL Multiphysics>PDE Modes> PDE, Coefficient Form>Eigenvalue analysis.
- 3 Verify in the Element list that Lagrange Quadratic is selected.
- 4 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu select **Constants**.
- 2 Enter the following constants, and when done, click OK.

NAME	EXPRESSION	
m	5.68e-12	
hbar	6.58e-16	
ec1	0.697	
ec2	0	
1	0	

Note that ec1 and ec2 are the step potentials in eV (assuming the electrons are free within the InAs quantum dot/wetting layer), and *l* is a quantum number.

3 From the Options menu choose Expressions>Scalar Expressions.

4 Enter the following expressions; when done, click **OK**:

NAME	EXPRESSION
A1	hbar^2/(2*m*0.067)
A2	hbar^2/(2*m*0.023)

- 5 From the Options menu select Axes/Grid Settings.
- 6 Enter the following axis definitions on the **Axis** page and the **Grid** page; on the **Grid** page you might need to clear the **Auto** check box. When done, click **OK**.

AXIS		GRID	
x min	- 5	x spacing	5
x max	30	Extra x	12
y min	- 60	y spacing	5
y max	60	Extra y	-1,1,3.6

GEOMETRY MODELING

I From the **Draw** menu select **Specify Objects>Rectangle**, then enter these settings; when done, click **OK**:

SIZE		POSITION	
Width	25	Base	Center
Height	100	x	12.5
		у	0

2 From the Draw menu select Specify Objects>Rectangle, then enter these settings; when done, click OK:

SIZE		POSITION	
Width	25	Base	Center
Height	2	x	12.5
		у	0

- **3** On the Draw toolbar click the **Line** button. Then draw a triangle with vertices at the coordinate pairs (0, 0), (12, 0), and (0, 3.6). Use the right mouse button to complete this object, which becomes **CO1**.
- 4 On the Draw toolbar click the **Create Composite Object** button. Select the objects **R2** and **C01**. Clear the **Keep interior boundaries** check box. Click **OK**.

- 5 On the Draw toolbar click the Create Composite Object button. Select the objects R1 and CO2, make sure the Keep interior boundaries check box is selected, then click OK.
- 6 On the Draw toolbar click the Scale button, then, in the Scale factor area, enter a scale factor of 1e-9 for both the x and y axes. Click OK.
- 7 Click the Zoom Extents button on the Main toolbar.

Boundary Conditions

- I From the Physics menu select Boundary Settings.
- **2** Enter the following boundary conditions:

SETTINGS	BOUNDARIES 2, 9	ALL OTHERS BOUNDARIES
Туре	Dirichlet	Neumann
q	0	0
g	0	0
h	1	
r	0	

3 When done, click **OK**.

Subdomain Settings

I From the Physics menu select Subdomain Settings.

2 Enter the PDE coefficients from this table and then click **OK**.

SETTINGS	SUBDOMAINS I, 3	SUBDOMAIN 2
с	A1	A2
a	A1/(x^2)*l*l+ec1	A2/(x^2)*l*l+ec2
β	-A1/x, 0	-A2/x, 0

MESH GENERATION

From the Mesh menu select Initialize Mesh.

COMPUTING THE SOLUTION

- I From the Solve menu select Solver Parameters.
- 2 On the General page, type 4 in the Desired number of eigenvalues edit field.
- 3 Click OK.
- 4 Click the **Solve** button.

POSTPROCESSING AND VISUALIZATION

- I From the **Postprocessing** menu select **Plot Parameters**.
- **2** On the **General** page find the **Solution to use** area, then go to the **Eigenvalue** list, and select the desired eigenvalue (keep in mind that these solutions are for a quantum number l = 0).
- **3** Click **Apply**.
- 4 Click the **Surface** tab, then click the **Height Data** tab and select the **Height data** check box.
- 5 Click Apply, then click OK.

12

Semiconductor Device Models

Semiconductor device modeling involve diffusion processes for hole and electron concentrations as well as electric potential distribution. Thermal effects are also a factor. Semiconductor models often include multiphysics couplings and have highly nonlinear material properties, making them difficult to solve.

Semiconductor Diode¹

Introduction

A *semiconductor diode* consists of two regions with different doping: a p-type region with a dominant concentration of holes, and an n-type region with a dominant concentration of electrons. The *anode* contact ties to the p-type region, and the *cathode* connects to the n-type region. Impurities that the manufacturing process adds to the semiconductor material determine each region's doping type. Scientists typically denote the dopant concentration with *N*. A negative value indicates acceptor atoms (*p*-type material), and a positive value indicates donor atoms (*n*-type material).

It is possible to derive a semiconductor model from Maxwell's equations and Boltzmann transport theory with the aid of some generally accepted simplifications such as assuming the absence of magnetic fields and the constant density of states. The model presented here formulates the problem using three dependent variables: ψ , n, and p. Even in this model's simplest form, strong nonlinear dependencies are present. The three basic semiconductor equations appear below (Ref. 1).

$$\begin{split} -\nabla \cdot (\varepsilon \nabla \Psi) &= q(p-n+N) \\ -\nabla \cdot \mathbf{J}_n &= -q R_{\mathrm{SRH}} \\ -\nabla \cdot \mathbf{J}_p &= q R_{\mathrm{SRH}} \end{split}$$

These equations contain the following variables:

- ψ indicates electrostatic potential
- q equals the elementary charge
- *p* and *n* are the hole and electron concentrations, respectively
- N represents the fixed charge associated with ionized donors

You can express the electron and hole current densities, \mathbf{J}_n and \mathbf{J}_p , in terms of ψ , n and p:

$$\begin{aligned} \mathbf{J}_n &= -qn\mu_n \nabla \psi + qD_n \nabla n \\ \mathbf{J}_p &= -qp\mu_p \nabla \psi - qD_p \nabla p \end{aligned}$$

where μ_n and μ_n are the carrier mobilities, and D_n and D_n are the carrier diffusivities.

^{1.} Model provided by Erik Danielsson, Royal Institute of Technology, Stockholm.

The term R_{SRH} represents the *Shockley-Read-Hall recombination*, which is a general recombination process using traps in the forbidden band gap of the semiconductor.

$$R_{\rm SRH} = \frac{np - n_i^2}{\tau_p(n+n_1) + \tau_n(p+p_1)}$$

Here n_i is the intrinsic carrier concentration, τ_n and τ_p are the carrier lifetimes, and n_1 and p_1 are parameters related to the trap energy level. If the trap level is located at the middle of the band gap (which the model assumes), then n_1 and p_1 equal n_i .

BOUNDARY CONDITIONS

For boundaries in contact with an insulator, you can use symmetric boundary conditions. For boundaries far away from the active device area, the electric field and carrier currents have zero normal component. You can describe both of these types with Neumann boundary conditions

$$\mathbf{n} \cdot \nabla \mathbf{\Psi} = \mathbf{0}$$
$$\mathbf{n} \cdot \mathbf{J}_n = \mathbf{0}$$
$$\mathbf{n} \cdot \mathbf{J}_p = \mathbf{0}$$

At boundaries in contact with a metal, the electrostatic potential is fixed. If you assume infinite recombination velocity at the contact, the mass action law is valid:

$$n_i^2 = np.$$

It is now possible to calculate the carrier concentrations using this law along with the dopant concentration N and the assumption that there is no charge at the contact. The applied voltage equals the Fermi level in the semiconductor at the contact, so the electrostatic potential at the contact is the applied voltage plus the energy difference between the Fermi level and the reference level used for the electrostatic potential. The energy difference has a logarithmic dependence of the carrier concentrations. The boundary conditions are

$$\begin{split} \Psi &= V_{\mathrm{a}} + \frac{kT}{q} \mathrm{ln} \left(\frac{\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2}}{n_i} \right) \\ n &= \frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2} \\ p &= -\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2} \end{split}$$

where V_a is the applied voltage. This is the simplest type of electrical contact. It is also possible to define a specific contact metal (some examples being aluminum or titanium) and to specify the current at the contact instead of the voltage.

Model Definition

DIODE GEOMETRY

The diode's geometry appears below.



Thicker lines indicate electrical contacts. This example applies a potential V_a to the anode, and the cathode is grounded.

VARIABLE	VALUE
У	7·10 ⁻⁶ m
×I	5·10 ⁻⁶ m
ju	I·I0 ⁻⁶ m
ac	2·10 ⁻⁶ m

Use the following notation for the dimensions of the device.

MESHING

The large dynamic range of the carrier concentrations makes the numerical simulation of the semiconductor equations a difficult task, especially because the entire range changes in the vicinity of the junction. Therefore you must select the mesh quite carefully, otherwise the solution process might not converge or could give inaccurate results. Thus, this example places a subdomain boundary at the junction to establish high accuracy. You must make the mesh denser around the junction and at the cathode and anode. Use the adaptive solver to obtain additional accuracy.

MATERIAL PROPERTIES

The most common material used for semiconductor devices is silicon, and the following table summarizes some key material properties used in this model.

NOTATION	VARIABLE	VALUE
n _i	ni	$1.46 \cdot 10^{10} \text{ cm}^{-3}$ (T = 300K)
ε _r	epsilonr	11.8
μ_n	mun	800 cm ² /Vs (low conc.)
μ_p	mup	200 cm ² /Vs (low conc.)
$D_n = \mu_n kT/q$	Dn	20.7 cm ² /s
$D_p = \mu_p kT/q$	Dp	5.17 cm ² /s
τ_n	taun	~0.1 µs
τ_p	taup	~0.1 µs

Mobilities and diffusivities often depend on N, but this effect is relatively small in the diode structure. The carrier lifetime can vary from a few picoseconds up to several microseconds and depends on parameters such as doping, temperature, and recombination centers.

DOPANT CONCENTRATION FUNCTION

The p-type doping of the diode can be accomplished by implantation or diffusion of acceptor atoms. The resulting profile from these two steps is often approximated with a gauss function. In addition, the transition between the n-type doped substrate and the lower doped n-type epitaxial layer can also be approximated with a Gauss function, because the dopants diffuse from the highly doped substrate during the high temperature step of the epitaxial growth. The final expression for the diode doping is

$$N = N_{Dn} + N_{Dn\max}G(x, -y, \infty, y_1) - N_{Ap\max}G(x, y, x_1, 0)$$

where N_{Dn} is the doping of the epitaxial layer, N_{Dnmax} the doping of the substrate, and N_{apmax} the peak of the implantation of diffusion profile. *G* is the Gauss function defined by

$$G(x, y, x_1, y_1) = \begin{cases} e^{-\frac{(x-x_1)^2}{ch^2}} e^{-\frac{(y-y_1)^2}{ch^2}} \\ e^{-\frac{(x-x_1)^2}{ch^2}} \\ e^{-\frac{(x-x_1)^2}{ch^2}} \\ e^{-\frac{(y-y_1)^2}{ch^2}} \\ e^{-\frac{(y-y_1)^2}{ch^2}} \\ 1 \\ x \le x_1 \\ y \le y_1 \end{cases}$$

The following constants are used in the expression for N.

NOTATION	VARIABLE	VALUE
N _{Dn}	NDn	10 ¹⁵ cm ⁻³
N _{Dnmax}	NDnmax	10 ¹⁷ cm ⁻³
N _{apmax}	NApmax	10 ¹⁷ cm ⁻³
<i>x</i> ₁	ac	2 µm
<i>y</i> ₁	yl	7 µm
ch	ch	calculated from ju

BOUNDARY CONDITIONS

The contact boundary conditions are

$$\begin{split} \psi &= V_{a} + \frac{kT}{q} \ln \left(\frac{\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^{2} + n_{i}^{2}}}{n_{i}} \right) \\ &n &= \frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^{2} + n_{i}^{2}} \\ &p &= -\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^{2} + n_{i}^{2}} \end{split}$$

To compute an accurate value of the current you can use weak constraints for the contact boundary conditions. To implement the weak boundary conditions on n and p you need to introduce Lagrange multipliers lm_2 and lm_3 on the anode and cathode. The variables lm2 and lm3 provide a more accurate value than what you obtain by the expression for the current densities. Therefore, the outward normal current densities are q*lm2 and -q*lm3, for electrons and holes, respectively.

PARAMETRIC STUDY

Solving the numerical problem for higher anode voltage is extremely difficult. You must increase the voltage from 0 V in small steps and use the solution from a previous step as the initial condition for the next one. The parametric solver takes care of this automatically. The anode voltage $V_{\rm a}$ varies from 0 V to 1 V in steps of 0.025 V.

You must also provide the nonlinear solver with an initial guess for the solution corresponding to $V_a = 0$ that is compatible with the boundary conditions.

$$\Psi = \frac{kT}{q} \ln \left(\frac{\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2}}{n_i} \right)$$
$$n = \frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2}$$
$$p = -\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2}$$

The initial guess for lm_2 and lm_3 is zero. Because the problem is nonlinear, you should convert it to general form to get a correct Jacobian and fast convergence.

COMSOL Multiphysics lets you scale the degrees of freedoms by setting reference values for the degrees of freedom. The relative tolerance in the nonlinear solver is by

default 10^{-6} . The absolute tolerance is the scaling times the relative tolerance. The solver disregards errors that are below the absolute tolerance. In this case the absolute tolerance for the variables ψ , u, and v is 10^{-6} , while the absolute tolerance for Im_2 and Im_3 is 10^{-4} . The absolute tolerance for Im_2 and Im_3 should be at least 10^{-4} , because the round-off errors in the equations induce errors of this magnitude in Im_2 and Im_3 . For this problem, the parameter step length control works best if you use the highly nonlinear option.

Results

When no bias is applied, the hole concentration is at thermal equilibrium. With 0.5 V forward bias, the holes from the p-type region flow into and through the low doped n-type region without significant recombination (see Figure 12-1). As a result, the hole concentration increases several orders of magnitude in the n-type region.



Figure 12-1: The hole concentration at 0.5 V forward bias.

COMPUTING THE IV CHARACTERISTICS

To compute the currents through the anode and cathode you have to integrate the normal component of the current densities

$$\mathbf{n} \cdot \mathbf{J}_n$$

 $\mathbf{n} \cdot \mathbf{J}_p$

which are more accurately available by the Lagrange multipliers lm_2 and lm_3 , respectively. The currents through the anode and cathode are

$$I_a = 10^{-6} \int_{\text{anode}} -q(\ln_u - \ln_2)$$
$$I_c = 10^{-6} \int_{\text{cathode}} q(\ln_u - \ln_3)$$

Because the geometry is 2D, the unit A/ μ m is used for the current. Note that I_a and I_c are vectors with one value for each value of the voltage. The exact agreement of the anode and cathode currents is a consequence of the finite element method. To see this, consider the weak equations

$$0 = \int_{\Omega} \left(\frac{R_{\text{SRH}}}{n_i} u_{\text{test}} + \mathbf{J}_n \cdot \nabla u_{\text{test}} \right) dA - \int_{B} \lim_{2} \cdot u_{\text{test}} \, ds$$
$$0 = \int_{\Omega} \left(\frac{R_{\text{SRH}}}{n_i} v_{\text{test}} + \mathbf{J}_p \cdot \nabla v_{\text{test}} \right) dA - \int_{B} \lim_{3} \cdot v_{\text{test}} \, ds$$

which are valid for all test functions u_{test} and v_{test} in the finite element space. *B* is the union of the anode and the cathode. Taking $u_{\text{test}} = v_{\text{test}} = 1$ and subtraction of these equations gives

$$\int_{B} (\ln_2 - \ln_3) ds = 0$$

which shows that $I_a = I_c$. The plot in Figure 12-2 shows the current versus the voltage.

Because the current for zero voltage is zero (theoretically), it has been omitted from the plot.



Figure 12-2: The IV characteristics of the semiconductor diode plotted in a logarithmic scale.

One very important parameter in the diode equation is the ideality factor, η .

$$I = I_0 \left(e^{\frac{qV}{\eta kT}} - 1 \right).$$

The factor η lies between 1 and 2, where $\eta \sim 1$ represents an ideal diode. In a diode with $\eta \sim 2$, the current is recombination/generation controlled (Ref. 1). In the IV-characteristic curve, you can identify two different slopes in the diode's forward bias region. These slopes are dependent on the ideality factor, the first with $\eta = 1.6$ and the second with $\eta = 1.08$. This situation is common for ordinary silicon diodes; at low forward biases the current is almost recombination controlled, and when the bias is increased the current characteristics becomes almost ideal ($\eta \sim 1$).

References

1. Sze, S.M., *Physics of Semiconductor Devices*, John Wiley & Sons, 1981, ISBN 0-471-05661-8.

2. Selberherr, S, *Analysis and Simulation of Semiconductor Devices*, Springer-Verlag, 1984, ISBN 0-387-81800-6.

Model Library path:

COMSOL_Multiphysics/Semiconductor_Devices/semiconductor_diode

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select 2D from the Space dimension list.
- 2 Click the **Multiphysics** button.
- 3 In the COMSOL Multiphysics>Electromagnetics folder, select Electrostatics. Make sure Lagrange Quadratic is selected in the Element list.
- 4 Enter psi in the Dependent variables edit field.
- 5 Click Add.
- 6 In the COMSOL Multiphysics>Convection and Diffusion folder, select Convection and Diffusion and then Steady-state analysis. Make sure Lagrange Quadratic is selected in the Element list.
- 7 Enter cn in the Dependent variables edit field.
- 8 Click Add.
- **9** Click the **Application Mode Properties** button, select **Conservative** from the **Equation** form list and **On** from the **Weak constraints** list. Click **OK**.
- 10 In the COMSOL Multiphysics>Convection and Diffusion folder, select Convection and Diffusion and then Steady-state analysis. Make sure Lagrange - Quadratic is selected in the Element list.
- II Enter cp in the Dependent variables edit field.
- 12 Click Add.
- I3 Click the Application Mode Properties button. Then select Conservative from the Equation form list and On from the Weak constraints list. Click OK.

I4 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- **2** In the **Constants** dialog box, define the following constants with names and expressions:.

NAME	EXPRESSION	DESCRIPTION
q	1.602e-19[C]	Elementary charge
Т	300[K]	Room temperature
k	1.38e-23[J/K]	Boltzmann's constant
epsilonr	11.8	Rel. permittivity for Si
ni	1.46e10[1/cm^3]	Intrinsic concentration for Si
mun	800[cm^2/(V*s)]	Electron mobility for Si
mup	200[cm^2/(V*s)]	Hole mobility for Si
Dn	k*T/q*mun	Electron diffusivity
Dp	k*T/q*mup	Hole diffusivity
taun	0.1[us]	Electron life time
taup	0.1[us]	Hole life time
С	q/(k*T)	Reciprocal thermal voltage
y1	7[um]	Diode dimension
x1	5[um]	Diode dimension
ju	1[um]	Junction depth
ac	2[um]	Anode dimension
NApmax	1e17[1/cm^3]	Maximum p-type doping
NDn	1e15[1/cm^3]	Drift layer n-type doping
NDnmax	1e17[1/cm^3]	Maximum n-type doping
ch	ju/sqrt(log(NApmax/NDn))	Doping fall-off constant

- 3 From the Options menu, choose Axes/Grid Settings.
- 4 Enter the axis and grid settings in the table below. To enter the grid spacing, first clear the **Auto** check box

AXIS		GRID	
x min	0	x spacing	1e-6
x max	1e-5	Extra x	

AXIS		GRID	
y min	-8e-6	y spacing	1e-6
y max	0	Extra y	

5 Click OK.

GEOMETRY MODELING

- I Draw a rectangle from (0, -7e-6) to (5e-6, 0).
- 2 Turn off the solidify feature by double-clicking the **SOLID** button on the status bar.
- **3** Click the Line button and draw a line from (0, -1e-6) to (2e-6, -1e-6).
- 4 Click the **2nd Degree Bézier Curve** button and continue with an arc to (3e-6, -1e-6) and finish the arc at (3e-6, 0).
- **5** Click the **Point** button and add a point at (2e-6, 0).
- 6 Click the Zoom Extents button on the Main toolbar.

PHYSICS SETTINGS

Expression Variables

- I From the Options menu, point to Expressions, and then click Scalar Expressions.
- 2 Enter each item in the table below as a single line without the line break.

NAME	EXPRESSION	DESCRIPTION
N	NDn+NDnmax*exp(-((y+y1)/ch)^2)- NApmax*exp(-(y/ch)^2)*((x <ac)+ (x>=ac)*exp(-((x-ac)/ch)^2))</ac)+ 	Doping concentration
RSRH	(cn*cp-ni^2)/(taup*(cn+ni)+ taun*(cp+ni))	Recombination term
psi_init	1/c*(-log(p_init/ni)*(N<0)+ log(n_init/ni)*(N>=0))	Charge neutrality voltage
n_init	<pre>(abs(N)/2+sqrt(N^2/4+ni^2))* (N>=0)+ni^2/(abs(N)/2+ sqrt(N^2/4+ni^2))*(N<0)</pre>	Charge neutrality electron concentration
p_init	<pre>(abs(N)/2+sqrt(N^2/4+ni^2))* (N<0)+ni^2/(abs(N)/2+ sqrt(N^2/4+ni^2))*(N>=0)</pre>	Charge neutrality hole concentration

3 Click **OK**. The inconsistent unit warning for the variable RSH is due to the use of mol as unit For the solution variables in the Convection and Diffusion application modes. The convention in Semiconductor physics is to use 1/cm³ for

concentrations, and this is the assumed unit for the solution variables. The units for solution variables cannot be changed by the user.

Integration Coupling Variables

The cathode current Ic is defined as an integral of the normal current density along the cathode. This can be implemented by using an integration coupling variable:

- I From the Options menu, point to Integration Coupling Variables and then click Boundary Variables.
- Select Boundary 2 and then type Ic in the Name column, 1e-6*q*(1m2-1m3) in the Expression column, and 4 in the Integration order column. Click OK.

Boundary Conditions

- I On the Multiphysics menu, select the Electrostatics (es) application mode.
- 2 Open the Boundary Settings dialog box and enter the following settings:

SETTINGS	BOUNDARIES 1, 3, 6, 7, 8	BOUNDARY 2	BOUNDARY 5
Boundary condition	Zero charge/Symmetry	Electric potential	Electric potential
V ₀		psi_init	Va+psi_init

- 3 On the Multiphysics menu, select the Convection and Diffusion (cd) application mode.
- 4 Open the Boundary Settings dialog box and enter the following settings:

SETTINGS	BOUNDARIES I, 3, 6-8	BOUNDARIES 2, 5
Boundary condition	Insulation/Symmetry	Concentration
cn ₀		n_init

- **5** On the **Multiphysics** menu, select the **Convection and Diffusion (cd2)** application mode.
- 6 Open the Boundary Settings dialog box and enter the following settings:

SETTINGS	BOUNDARIES 1, 3, 6, 7, 8	BOUNDARIES 2, 5
Boundary condition	Insulation/Symmetry	Concentration
cp ₀		p_init

Subdomain Settings

I On the Multiphysics menu, select the Electrostatics (es) application mode.
2 Open the Subdomain Settings dialog box and enter the following settings:

SETTINGS	SUBDOMAINS I, 2	
ϵ_r (isotropic)	epsilonr	
ρ	q*(N-cn+cp)	

3 Click the **Init** tab, and enter the following initial value:

SETTING	SUBDOMAINS 1, 2
psi(t ₀)	psi_init

4 On the Multiphysics menu, select the Convection and Diffusion (cd) application mode.

5 Open the Subdomain Settings dialog box and enter the following settings

SETTINGS	SUBDOMAINS 1, 2	
D (isotropic)	Dn	
u	mun*psix	
v	mun*psiy	
R	- RSRH	

6 Click the **Init** tab and enter the following initial value:

SETTING	SUBDOMAINS 1, 2
cn(t ₀)	n_init

- 7 On the Multiphysics menu, select the Convection and Diffusion (cd2) application mode.
- 8 Open the Subdomain Settings dialog box and enter the following settings:

SETTINGS	SUBDOMAINS 1, 2	
D (isotropic)	Dp	
u	-mup*psix	
v	-mup*psiy	
R	-RSRH	

9 Click the **Init** tab, and enter the following initial value:

SETTINGS	SUBDOMAINS 1, 2
cp(t ₀)	p_init

MESH GENERATION

- I From the Mesh menu, choose Free Mesh Parameters.
- 2 Click the **Boundary** tab.
- **3** Select Boundaries 2, 4, 5, and 9 and type 1e-7 in the Maximum element size edit field.
- 4 Click the **Remesh** button and then click **OK**.

COMPUTING THE SOLUTION

- I From the Solve menu, choose Solver Parameters.
- 2 In the Solver list, select Parametric.
- **3** Type Va in the **Parameter name** edit field and 0:0.025:1 in the **Parameter values** edit field.
- 4 Click the Stationary tab. Enter 50 in the Maximum number of iterations edit field.
- 5 Select the Highly nonlinear problem check box.
- 6 Click **OK** and then click the **Solve** button to start the simulation.

POSTPROCESSING AND VISUALIZATION

The plot visualizes the logarithm of the hole concentration for the parameter value, $V_a = 0.5$ V.

- I Open the Plot Parameters dialog box.
- 2 On the General page, select 0.5 from the Parameter value list.
- 3 Click the Surface tab. Enter the expression log10(cp) in the Expression edit fields on the Surface Data tab and the Height Data tab. On the Height Data tab, select the Height data check box.
- 4 Click **OK** to get the plot shown in Figure 12-1 on page 448.
- 5 To plot the IV characteristics open the Domain Plot Parameters dialog box.
- 6 On the General page, click the **Point plot** button and select all parameters except 0 from the **Solutions to use** list.
- 7 Click the Title/Axis button to open the Title/Axis Settings dialog box. Select the Log scale check box for the secondary axis. Click OK.
- 8 On the **Point** tab, type Ic in the **Expression** edit field.
- **9** Select Vertex 1 and click **OK**. As a separate window you see the plot in Figure 12-2 on page 450.

Distributed SPICE Model of an Integrated Bipolar Transistor

Introduction

Integrated semiconductor devices are used in almost all electrical equipment, and the bipolar junction transistor (BJT) is still a very important device although the MOS technology has taken over a large part of the market. Especially in technologies with new types of semiconductors the bipolar transistor can be the best choice for optimum performance. The BJT consists of three semiconductor layers of alternating dopant type (N-type and P-type) forming an NPN or a PNP structure. These structures can control a large current using a much smaller current.

This particular example describes a high-voltage NPN BJT fabricated from a semiconductor called silicon carbide (SiC). This semiconductor device combines high voltage properties (> 1000 V) and fast switching (> 1 MHz), which is impossible to achieve with comparable silicon devices. The applications for the device are in the drive electronics of electric motors and switched voltage supplies. A higher frequency decreases the size of the passive devices, like inductors and capacitors.

Due to the complicated system of PDEs necessary to calculate the behavior of a semiconductor device, 3D simulations are rather impractical. As a result, most such simulations are performed in 2D, and much effort is put into finding symmetries and removing inactive areas. However, several design considerations need a calculation including the full layout of the device, especially in cases where significant voltage drops from high currents are expected in the metal layers. Further simplifications are therefore necessary, and a powerful approach for BJT structures is to use distributed SPICE expressions on a 2D top view layout of the device.



Figure 12-3: 3D view over the layers included in the model. The actual simulation is in 2D, using the top view of this figure.

Model Definition

The potential in each layer obeys the equation

$$\nabla \cdot \left(\frac{W_n}{\rho_n} \nabla V_n\right) = \sum_{m \neq n} J_{n \to m}$$

where W_n is the layer thickness, ρ_n the layer resistivity, and V_n the potential of layer n. The right side is just the sum of all currents going from layer *n* to layer *m*. These currents are calculated using SPICE expressions, which is a function on the layer potentials at each point (see Table 12-1 on page 459).

The boundary condition for all boundaries sets the outward current flow to zero.

The electrical connection to the BJT is modeled through point conditions to the base metal and the emitter metal layers. The entire collector layer is connected to a fixed applied voltage through a distributed resistance (models a backside contact). The point conditions can either model a probe tip in on-chip measurements or a bond wire in a packaged device. The emitter point condition is just a constraint to zero volts for the dependent variable of this layer. The base contact condition can either be a constraint on the applied voltage, or a constant current condition. For the latter condition the model uses a point source.

Table 12-1 presents the different SPICE devices along with the expression for each device. Because the devices are distributed, the unit is per area where applicable.

TABLE 12-1: SPICE DEVICES IN THE MODEL

SPICE MODEL	EXPRESSION
Resistor	$J_{n \to m} = (V_n - V_m) / R_{nm}$
Capacitor (trans)	$J_{n \to m} = C_{nm} \frac{d}{dt} (V_n - V_m)$
Capacitor (ac)	$J_{n \to m} = j \omega C_{nm} (V_n - V_m)$
ВЈТ	$J_{b \to e} = \frac{I_{be1}}{B_F} + I_{be2}$ $J_{b \to c} = I_{bc2}$ $J_{c \to e} = \frac{I_{be1}}{K_{qb}}$

Note the following about the expressions above:

- The subscripts *n* and *m* denote one of the subscripts *e*, *b*, *c*, or *m*, depending on the layer the expression refers to.
- The current always has a counterpart representing the flow in the opposite direction.

$$J_{n \to m} = -J_{m \to n}$$

The variables not defined yet are SPICE parameters, which you define as constants in the COMSOL Multiphysics model. The SPICE expressions come from the Gummel-Poon model of the bipolar transistor with a few simplifications (Ref. 1). Because the distributed SPICE model is a transistor with no lateral extension, all parameters referring to lateral losses are ignored (the top view geometry takes the lateral part into account). The voltage dependence on junction capacitance is ignored along with some time constants, because the model is already quite complex. The parameter values are fitted to measurements and device simulations on a silicon carbide BJT fabricated at KTH (The Royal Institute of Technology), Stockholm, Sweden.

Results and Discussion

Figure 12-4 shows the base and collector currents as a function of time when the base is ramped from 2 V to 4 V in 100 ns at a constant collector voltage of 15 V. The figure

reveals that the transistor has no problem to follow the ramp, and only a small overshoot in the base current can be detected after 100 ns.

0.05 0.04





Ic

Figure 12-4: Turn on in 100 ns, where the red curve is the base current and the blue curve is the collector current.

Figure 12-5: Turn on in 10 ns, and now the transistor cannot follow the ramp.

When the rise time is decreased to 10 ns the BJT runs into problems, and Figure 12-5 shows that the actual turn on time is about 30 ns. If faster ramps are required the simulation results have the information about the details on the limiting factors, but in some cases an harmonic simulation is easier to interpret. The plot in Figure 12-6 shows the harmonic component of the collector current density, taken from a harmonic simulation performed at a bias point of 10 mA base current, 15 V collector voltage, and at a 100 MHz frequency. This result shows a significant density beneath the base pad, which does not take part in the current amplification. An improved design would be to use a second metal layer for the pads, resulting in a more efficiently used device area.



Figure 12-6: The harmonic part of the collector current density at 100 MHz.

These results only qualitatively estimate the switch speed of the device. In an experimental setup, other parasitic elements influence the result (for example, probes, package, and cables). The approximation on the capacitances also affects the result.

Modeling in COMSOL Multiphysics

Use four PDE, General Form application modes—one for each layer—to group the voltages. Within each application mode there are two dependent variables, one for the DC voltage and one for the AC voltage (complex valued).

To model the point current source, use a test function and the weak form settings.

Reference

1. H. K. Gummel and H.C. Poon, "An Integral Charge Control Model of Bipolar Transistors," *The Bell System Technical Journal*, vol. 49, pp. 827–852, 1970.

Model Library path: COMSOL_Multiphysics/Semiconductor_Devices/ bipolar_transistor

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Click the Multiphysics button in the Model Navigator.
- 2 Select 2D from the Space dimension list.
- 3 Select the COMSOL Multiphysics>PDE Modes>PDE, General Form application mode. In the Dependent variables edit field, enter the names of the variables: Vm_dc Vm_ac. Enter metal as the application mode name, and select the Lagrange - Linear element. Click the Add button to add the mode to the model.
- 4 Add the second application mode of the same type and with the same element. Use the variable names: Ve_dc Ve_ac. Enter emitter as the application mode name.
- 5 Add the third application mode of the same type and with the same element. Use the variable names: Vb_dc Vb_ac. Enter base as the application mode name.
- 6 And finally, add the fourth application mode of the same type and with the same element. Use the variable names: Vc_dc Vc_ac. Enter collector as the application mode name.
- 7 Click OK.

OPTIONS AND SETTINGS

I In the **Constants** dialog box, enter the following variable names and expressions. The variable t does not affect the time-dependent simulation, because the model uses the internal time variable.

NAME	EXPRESSION	DESCRIPTION
Vbe	2.0	Applied base-emitter voltage (V)
Vbe_on	4.0	Final voltage in transient simulations (V)
t_rise	10e-9	Rise time to reach Vbe_on (s)
Vce	15	Applied collector-emitter voltage (V)
Vac	1	Applied AC voltage (V)
Ib_a	0.0	Base current forced into base contact (A)

NAME	EXPRESSION	DESCRIPTION	
Vt	0.0259	Thermal voltage	
freq	10e6	Frequency (Hz)	
omega	2*pi*freq	Angular frequency (rad/s)	
Wb	0.3e-6	Thickness of base layer (m)	
We	0.5e-6	Thickness of emitter layer (m)	
Ws	0.5e-6	Thickness of sub-collector layer (m)	
Wm	0.2e-6	Thickness of metal layer (m)	
rhoB0	4.54e-3	Max resistivity of base layer (Ω ·m)	
rhoBm	0.783e-3	Min resistivity of base layer ($\Omega \cdot m$)	
rhoE	0.293e-3	Resistivity of emitter layer (Ω ·m)	
rhoS	0.290e-3	Resistivity of sub-collector layer (Ω ·m)	
rhoM	1e-6	Resistivity of metal layer (Ω ·m)	
Cbe	3.29e-3	Base-emitter capacitance (F/m ²)	
Cbc	3.38e-3	Base-collector capacitance (F/m ²)	
Rbm	1e-9	Resistance between metal and base layer ($\Omega \cdot m^2$)	
Rem	1e-10	Resistance between metal and emitter layer (Ω ·m ²)	
Rs	7e-8	Resistance between the collector layer and the backside $(\Omega {\cdot} m^2)$	
Nf	1.0	Base-emitter ideality factor	
Ne	1.2	Base-emitter second ideality factor	
Nc	1.0	Base-collector ideality factor	
Nk	0.867	Beta roll-off power factor	
Vaf	979	Forward Early voltage (V)	
Bf	27.0	Forward current gain	
Is	9.37e-42	Base-emitter saturation current (A/m ²)	
Isc	1.91e-42	Base-collector saturation current (A/m ²)	
Ise	9.7e-35	Base-emitter second saturation current (A/m ²)	
Ikf	2.98e7	Beta roll-off current (A/m ²)	
t	0.0	Time for stationary solutions	

2 In the Scalar Expressions dialog box, enter the following expressions.

NAME	EXPRESSION	DESCRIPTION
Ibc2	<pre>Isc*exp((Vb-Vc)/(Nc*Vt))</pre>	Base-collector current (A·m ²)
Kq1	1/(1-(Vb-Vc)/Vaf)	Voltage dependent factor

NAME	EXPRESSION	DESCRIPTION	
Kq2	Ibe1/Ikf	Current dependent factor	
Kqb	Kq1/2*(1+(1+4*Kq2)^Nk)	Current and voltage dependent factor	
rhoB	rhoBm+(rhoBO-rhoBm)/Kqb	Current dependent base resistivity $(\Omega \cdot m^2)$	
Vbe_t	<pre>Vbe+(t/t_rise*(t<t_rise)+ (t="">=t_rise))*(Vbe_on-Vbe)</t_rise)+></pre>	Ramping of Vbe to final voltage (V)	
Vm	Vm_ac+Vm_dc	Total metal layer potential (V)	
Vmx	Vm_acx+Vm_dcx	X-derivative of layer potential (V/m)	
Vmy	Vm_acy+Vm_dcy	Y-derivative of layer potential (V/m)	
Ve	Ve_ac+Ve_dc	Total emitter layer potential (V)	
Vex	Ve_acx+Ve_dcx	X-derivative of layer potential (V/m)	
Vey	Ve_acy+Ve_dcy	Y-derivative of layer potential (V/m)	
Vb	Vb_ac+Vb_dc	Total base layer potential (V)	
Vbx	Vb_acx+Vb_dcx	X-derivative of layer potential (V/m)	
Vby	Vb_acy+Vb_dcy	Y-derivative of layer potential (V/m)	
Vc	Vc_ac+Vc_dc	Total collector layer potential (V)	
Vcx	Vc_acx+Vc_dcx	X-derivative of layer potential (V/m)	
Vcy	Vc_acy+Vc_dcy	Y-derivative of layer potential (V/m)	
Jbc_ac	j*omega*Cbc*(Vb_ac-Vc_ac)		

GEOMETRY MODELING

The geometry is constructed from a layout drawing in the GDS file format, which is the most common format used in the fabrication process of reticles for lithography equipment. The file format can be converted to a DXF file and imported to COMSOL Multiphysics. The spatial unit in the GDS file are given in nanometers (nm). Because the length unit in COMSOL Multiphysics is meters and not nanometers, you must scale the geometry.

- I In the File menu, select Import and choose CAD Data From File. Click Browse and select the file bipolar_transistor.dxf, which you find in the models/ COMSOL_Multiphysics/Semiconductor_Devices folder. Click Import.
- 2 Click the **Coerce to Solid** button on the Draw toolbar to make it a solid geometry.
- 3 On the **Draw** menu, point to **Modify**, and then click **Scale**. In the **x** and **y** scale factor edit fields, type the value 1e-9. Click **OK**.
- 4 Click the **Zoom Extents** button on the Main toolbar.

- 5 Choose **Draw>Specify Objects>Point**. In the **Point** dialog box, type 70e-6 in the **x** edit field and 220e-6 in the **y** edit field in the **Coordinates** area. Click **OK**. This defines the point PT1 at the location of the emitter.
- 6 Choose Draw>Specify Objects>Point. In the Point dialog box, type 260e-6 in the x edit field and 70e-6 in the y edit field in the Coordinates area. Click OK. This defines the point PT2 at the location of the base.



PHYSICS SETTINGS

To simplify the expressions in the subdomains and in the postprocessing, do the following steps:

I In the **Subdomain Expressions** dialog box, type the following expressions for the given subdomains.

NAME	EXPRESSION	SUBDOMAIN	DESCRIPTION
Ibe1	<pre>Is*exp((Vb-Ve)/(Nf*Vt))</pre>	4, 5	Base-emitter current
Ibe2	<pre>Ise*exp((Vb-Ve)/(Ne*Vt))</pre>	4, 5	2nd base-emitter current
Jbe_ac	j*omega*Cbe*(Vb_ac-Ve_ac)	4, 5	Harmonic base-emitter current

- **2** Select Subdomains 1–3, and the enter 0 in all three edit fields in the **Expression** columns to make all expression variables equal to zero in those subdomains.
- **3** In the **Subdomain Integration Variables** dialog box, type the expressions given in the table below.

NAME	EXPRESSION	SUBDOMAIN	DESCRIPTION
Ib	(Vm-Vb)/Rbm	2	Base current
Ic	(Vce-Vc)/Rs	all	Collector current
Ie	(Vm-Ve)/Rem	5	Emitter current

Boundary Conditions

All boundaries should be of Neumann type with all parameters set to zero.

- I Select the metal application mode from the Multiphysics menu.
- 2 Open the Boundary Settings dialog box.
- 3 Select all boundaries and select the Neumann boundary condition.
- **4** Select each of the other modes from the **Multiphysics** menu and do the same operations there.

Subdomain Settings (Point Settings, Edge Settings)

There are five subdomains with somewhat different settings.

- I Select the **metal** application mode from the **Multiphysics** menu.
- 2 Open the Subdomain Settings dialog box.
- 3 Select Subdomains 1, 3, and 4 and then clear the Active in this domain check box.
- **4** Enter the expressions for the Γ , *F*, and d_a coefficients according to the three tables below and click **OK**.

SETTINGS	SUBDOMAINS 2, 5	
Γ _I	Wm/rhoM*Vmx	Wm/rhoM*Vmy
Γ2	Wm/rhoM*Vmx	Wm/rhoM*Vmy
SETTINGS	SUBDOMAIN 2	SUBDOMAIN 5
F _I	(Vm-Vb)/Rbm	(Vm-Ve)/Rem
F ₂	(Vm-Vb)/Rbm	(Vm-Ve)/Rem

SETTINGS	SUBDOMA	INS 2, 5
d _{a, I}	0	0
d _{a,2}	0	0

5 Select the **emitter** application mode from the **Multiphysics** menu.

- 6 Open the Subdomain Settings dialog box, select Subdomains 1–3 and clear the Active in this domain check box.
- 7 Enter the expressions for the other subdomains according to the tables below and click **OK**.

SETTINGS	SUBDOMAINS 4, 5	
Γ ₁	We/rhoE*Vex	We/rhoE*Vey
Γ2	We/rhoE*Vex	We/rhoE*Vey

SETTINGS	SUBDOMAIN 4	SUBDOMAIN 5
F	-(Ibe1/Bf+Ibe2+Ibe1/Kqb)	(Ve-Vm)/Rem-(Ibe1/Bf +Ibe2+Ibe1/Kqb)
F ₂	-(Ibe1/Bf+Ibe2+Ibe1/Kqb) -Jbe_ac	(Ve-Vm)/Rem-(Ibe1/Bf +Ibe2+Ibe1/Kqb)-Jbe_ac

SETTINGS	SUBDOMAINS 4, 5	
d _{a, I}	-Cbe	0
$d_{a,2}$	0	0

- 8 Select the **base** application mode from the **Multiphysics** menu.
- 9 Open the Subdomain Settings dialog box. Enter the expressions for the subdomains according to the tables below, and click **OK**.

SETTINGS	ALL SUBDOMAINS	
Γ _I	Wb/rhoB*Vbx	Wb/rhoB*Vby
Γ ₂	Wb/rhoB*Vbx	Wb/rhoB*Vby

SETTINGS	SUBDOMAINS 1, 3	SUBDOMAIN 2	SUBDOMAINS 4, 5
FI	Ibc2	Ibc2+(Vb-Vm)/Rbm	Ibc2+Ibe1/Bf+Ibe2
F ₂	Ibc2+Jbc_ac	Ibc2+(Vb-Vm)/Rbm+Jbc_ac	Ibc2+Ibe1/Bf+Ibe2+ Jbe_ac+Jbc_ac

SETTINGS SUBDOMAINS I-3		SUBDOMAINS 4, 5		
d _{a, I}	-Cbc	0	-Cbe-Cbc	0
d _{a,2}	0	0	0	0

IO Select the **collector** application mode from the **Multiphysics** menu.

II Open the **Subdomain Settings** dialog box. Enter the expressions for the subdomains according to the tables below, and click **OK**.

SETTINGS	ALL SUBDOMAINS	
Γ _I	Ws/rhoS*Vcx	Ws/rhoS*Vcy
Γ2	Ws/rhoS*Vcx	Ws/rhoS*Vcy

SETTINGS	UBDOMAINS I-3	SUBDOMAINS 4,5
Fl	-Ibc2+(Vc-Vce)/Rs	-Ibc2+(Vc-Vce)/Rs+Ibe1/Kqb
F ₂	-Ibc2+(Vc-Vce)/Rs-Jbc_ac	-Ibc2+(Vc-Vce)/Rs+Ibe1/Kqb -Jbc_ac

SETTINGS	ALL SUBDOMAINS	
d _{a, I}	-Cbc	0
$d_{a,2}$	0	0

You must also specify the settings at the two points representing the probe or bond wire connections.

- I Select the **metal** application mode from the **Multiphysics** menu.
- 2 Open the Point Settings dialog box.
- **3** Enter the point expressions according to the table below.

SETTINGS	POINT 131	POINT 412
weak term	0 0	-Ib_a*Vm_dc_test 0
constraint	Vm_dc Vm_ac	(Vbe_t-Vm_dc)*(Ib_a==0) Vac-Vm_ac

Finally some cross time-derivative terms have to be specified, which you cannot specify from the **Subdomain Settings** dialog box. Use the equation-system settings for this.

I Open the Subdomain Settings - Equation System dialog box by selecting Physics>Equation System>Subdomain Settings.

Click the d_a tab and select Subdomains 1–3. Enter the fields specified in the table below. Leave all other existing expressions as is.

ROW NUMBER	COLUMN NUMBER	EXPRESSION
5	7	Cbc
7	5	Cbc

3 Select Subdomains 4 and 5 and fill in the fields specified below.

ROW NUMBER	COLUMN NUMBER	EXPRESSION
3	5	Cbe
5	3	Cbe
5	7	Cbc
7	5	Cbc

4 Click OK. Now this dialog box takes responsibility for the subdomain settings, which means that the software ignores all changes in the Subdomain Settings dialog box from now on. COMSOL Multiphysics indicates this with a padlock symbol next to all subdomain numbers. By clicking the Reset Equation button in the Subdomain Settings - Equation System dialog box, you set all parameters according to the Subdomain Settings dialog box and it is possible to do changes there again.

bdomain selection	c a f	e _a d _a o	β γ Ιηί	it Element	Weak \	ariables			
^	Damping	Mass coeff	icient						
	Vm_dc	Vm_ac	Ve_dc	Ve_ac	Vb_dc	Vb_ac	Vc_dc	Vc_ac	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	-Cbe	0	Cbe	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	Cbe	0	-Cbe-Cbc	0	Cbc	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	Cbc	0	-Cbc	0	
	0	0	0	0	0	0	0	0	
Select by group Reset Equation									

Initial Conditions

To speed up the DC simulation, the initial conditions of the metal, base, and collector layer can be set equal to the constant variables, Vbe and Vce. Due to the changes in the **Subdomain Settings - Equation System** dialog box, you must specify all the initial conditions there.

- I Open the Subdomain Settings Equation System dialog box, and click the Init tab.
- 2 Select Subdomain 2 and type Vbe in the Vm_dc(t₀) edit field.
- 3 Select all subdomains and type Vbe in the Vb_dc(t₀) edit field and Vce in the Vc_dc(t₀) edit field.
- 4 Click OK.

MESH GENERATION

Due to the small features in the defined geometry, it is possible to make the mesh coarser than the default.

- I In the Mesh menu, open the Free Mesh Parameters dialog box.
- 2 Select the Coarse predefined mesh size, and click the Remesh button.
- **3** Click the **Boundary** tab, select all boundaries that is part of the finger structure of the emitter and base, and enter 4e-6 in the **Maximum element size** edit field.
- 4 Click OK.

COMPUTING THE SOLUTION (TIME-DEPENDENT)

This sequence first finds the initial solution and then continue with the time-dependent simulation.

- I Open the Solver Manager dialog box.
- 2 Click the **Initial value expression** button in the **Initial value** area.
- Click the Solve For tab and expand all modes. Select all DC variables (ending with _dc). This makes the solver leave the AC application mode variables unchanged. Click OK.
- 4 Click the Solve button on the Main toolbar.
- 5 When the solution has been found, open the Solver Manager dialog box again.
- 6 Click the Initial Value tab. Click the Store Solution button, and then click the Stored solution button in the Initial value area to use the stored solution as the initial value. Click OK.

- 7 Open the Solver Parameters dialog box, and select the Time dependent solver. In the Times edit field, type 0 1e-6. To avoid large time steps, type 1e-3 in the Relative tolerance edit field and 5e-4 in the Absolute tolerance edit field.
- 8 Click the Time Stepping tab, and select Time steps from solver from the Times to store in output list.
- 9 Click OK, then click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION (TIME-DEPENDENT)

Use the defined coupling variables to plot the base and collector currents versus time:

- I In the Postprocessing menu, select Domain Plot Parameters.
- 2 Under the General tab select Point plot. In the Solutions to use area, select Interpolated times from the Select via list. In the Times edit field, type linspace(0,1e-7,100).
- **3** Click the **Point** tab and type Ib in the **Expression** edit field. Select point number 412 (for example by clicking on the point in the middle of the base pad).
- 4 Click the Line Settings button. In the Line Settings dialog box, select Color from the Line color list, and select Triangle from the Line marker list. Click the Color button and select red color from the palette. Click OK.



- 5 In the Domain Plot Parameters dialog box, click Apply to view the base current.
- 6 Click the General tab and select the Keep current plot check box.
- 7 Go back to the **Point** tab and type Ic in the expression edit field.

8 Change the color to blue and click the **OK** button. You now get the figure below, which shows how the currents behave during turn on.



COMPUTING THE SOLUTION (TIME-HARMONIC)

This sequence first finds a stationary solution for a base current of 10 mA and then uses this solution as the linearization point for the time-harmonic simulation:

- Open the Constants dialog box from the Options menu. Change the variable Vbe to
 2.5. Click OK. This gives a better initial condition for the constant current solution.
- 2 Open the Solver Parameters dialog box. Select the Parametric in the Solver list.
- **3** Type Ib_a in the **Parameter name** edit field and 1e-2 in the **Parameter values** edit field. Click **OK**.
- **4** Open the **Solver Manager** dialog box and select the **Initial value expression** option in the **Initial value** area. Click **OK**.
- **5** Click the **Solve** button.
- 6 When the analysis has finished, open the Solver Manager dialog box again.
- 7 Click the **Store Solution** button, and select the solution for the value of 0.01 in the **Parameter value** list. In the **Values of variables not solved for and linearization point** area, click the **Stored solution** option button.

- 8 Under the Solve For tab, expand all modes, deselect all dc variables and select all ac variables (ending with _ac). Click OK.
- 9 Open the Solver Parameters dialog box, and select Parametric in the Solver list.
- 10 Enter freq in the Parameter name edit field, and enter the values 1e6 1e7 1e8 in the Parameter values edit field.
- **II** Click the **Stationary** tab. Select **Linear** in the **Linearity** list to ensure that the parametric sweep uses the linear solver.
- I2 Click OK.
- **I3** Click the **Solve** toolbar button.

POSTPROCESSING AND VISUALIZATION (TIME-HARMONIC)

- I Open the Plot Parameters dialog box.
- 2 Click the Surface tab and make sure that the Surface plot and (on the Height Data tab)
 Height data check boxes are selected. Type abs((Vce-Vc)/Rs) in the Expression edit fields on the Surface Data tab and Height Data tab. Click OK.
- **3** Click the **Headlight** button on the Camera toolbar to the left to make the plot look nicer. The plot should now look like the following figure. Notice the high collector current density under the base pad.



Min: 4610.454

DC Characteristics of a MOS Transistor (MOSFET)

This model calculates the DC characteristics of a MOS (metal-oxide semiconductor) transistor using standard semiconductor physics. In normal operation, a system turns on a MOS transistor by applying a voltage to the gate electrode. When the voltage on the drain increases, the drain current also increases until it reaches saturation. The saturation current depends on the gate voltage.

Introduction

The MOSFET (Metal Oxide Semiconductor Field-Effect Transistor) is by far the most common semiconductor device, and the primary building block in all commercial processors, memories, and digital integrated circuits. During the past decades this device has experienced tremendous development, and today it is being manufactured with feature sizes of 90 nm and smaller.



Cross-section TEM (transmission electron microscope) image of a 70-nm MOSFET fabricated in the clean room at the Royal Institute of Technology in Kista, Sweden (a project of P.-E. Hellström and others).

This model shows the basic functionality of a MOS transistor, where the gate voltage controls the drain-source resistance and thus the drain current. At a certain gate voltage, V_{GS} , and at low drain voltages, the drain current is almost linearly dependent

on the drain voltage. When the drain voltage increases, the drain current saturates. The level of saturation depends on the gate voltage.

The gate voltage also influences the coefficient describing the linear dependence between the gate voltage and gate current at low drain voltages. This coefficient is generally known as the On resistance (R_{on}). In digital devices the transistor operates as a switch, making R_{on} an important parameter influencing the transistor's power loss and driving ability. Its ability to drive an output is of special importance. This characteristic is described by the *fanout*, the number of outputs that a device can drive. In high-speed systems, parasitic effects make the situation more complicated.

Model Definition

DEVICE GEOMETRY

The structure in Figure 12-7 is a cross section of a simplified MOS transistor. The electric field in the gate influences the low-doped p-type silicon; specifically, at a certain gate voltage a thin layer of it, close to the silicon-oxide surface, turns into an n-type material. This process, called inversion, creates a conducting channel between the highly doped n-type source and the drain regions. With this channel present, a voltage across the source and the drain drives a drain current. This model uses the following dimensions: a gate length of 0.2 μ m, a gate oxide thickness of 5 nm, and a source-drain junction depth of 0.1 μ m.



Figure 12-7: The model geometry. The simulation replaces the contacts with boundary conditions.

DOMAIN EQUATIONS

This example models a MOS transistor using the standard drift-diffusion approximation coupled with Poisson's equation. With the aid of some generally accepted simplifications—such as neglecting magnetic fields, assuming a constant density of states in both the valence and the conductance bands, and assuming a Boltzmann distribution of the carriers—it is possible to derive these equations from Maxwell's equations and Boltzmann transport theory.

The model in this example formulates the problem using three dependent variables: ψ (the electrostatic potential), *n*, and *p*. The three basic semiconductor equations are

$$\begin{split} -\nabla\cdot(\varepsilon\nabla\psi) &= q(p-n+N)\\ -\nabla\cdot\mathbf{J}_n &= -qR_{\mathrm{SRH}}\\ -\nabla\cdot\mathbf{J}_p &= qR_{\mathrm{SRH}} \end{split}$$

where p and n are the hole and electron concentrations, respectively, and N represents the fixed charge associated with ionized donors.

You can express the electron and hole current densities, J_n and J_p , with a drift term and a diffusion term

$$\begin{aligned} \mathbf{J}_{n} &= -qn\mu_{n}\nabla \psi + qD_{n}\nabla n\\ \mathbf{J}_{p} &= -qp\mu_{p}\nabla \psi - qD_{p}\nabla p \end{aligned}$$

where μ_n and μ_p are the carrier mobilities, and D_n and D_p are the carrier diffusivities.

Another term, R_{SRH} , represents the *Shockley-Read-Hall recombination*, which is a general recombination process using traps constituting energy levels in a semiconductor's forbidden band gap. In the recombination equation

$$R_{\rm SRH} = \frac{np - n_i^2}{\tau_p(n+n_1) + \tau_n(p+p_1)}$$

 n_i is the intrinsic carrier concentration, τ_n and τ_p are the carrier lifetimes, and n_1 and p_1 are parameters related to the trap energy level. If that level is located in the middle of the band gap (which this model assumes), then n_1 and p_1 equal the intrinsic carrier concentration, n_i .

BOUNDARY CONDITIONS

For boundaries in contact with an insulator or far away from the active device area, you can use the symmetry or zero charge (flux) boundary condition for the electrostatic (diffusion) problem. At boundaries in contact with a metal, the electrostatic potential is fixed. Assuming infinite recombination velocity at the contact, the mass action law

$$n_i^2 = np$$

is valid. Using this law along with the assumption that there is no charge at the contact, it is possible to calculate the carrier concentrations. The applied voltage equals the Fermi level in the semiconductor at the contact, so the electrostatic potential at the contact is the applied voltage plus the potential difference between the Fermi level and the electrostatic reference level:

$$\begin{split} \psi &= V_a - \chi_{Si} - \frac{E_G}{2} + \frac{kT}{q} \ln \left(\frac{\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2}}{n_i} \right) \\ & n &= \frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2} \\ & p &= -\frac{N}{2} + \sqrt{\left(\frac{N}{2}\right)^2 + n_i^2} \end{split}$$

where V_a is the applied voltage. In this model the reference potential is the vacuum level. This is often a good choice when several materials with different affinities or work functions are present. As a result, the electrostatic potential is continuous across all material interfaces. This reference level also finds use in heterojunction device simulations. The band diagram in Figure 12-8 shows how this model represents the metal-oxide semiconductor material system.



Figure 12-8: A band diagram of the MOS material system. The necessary parameters to align the materials are the affinities of the oxide and semiconductor, plus the work function of the polysilicon gate, which this model assumes has the same properties as aluminum.

NOTATION	VARIABLE	VALUE
$n_i(Si)$	ni_Si	$1.46 \cdot 10^{10} \text{ cm}^{-3}$ (T = 300K)
$\epsilon_r(Si)$	epsilonr_Si	11.8
$\epsilon_r(SiO2)$	epsilonr_Si	4.2
$\mu_n(Si)$	mun	1000 cm ² /Vs (low concentration)
$\mu_p(Si)$	mup	500 cm ² /Vs (low concentration)
$D_n = \mu_n(\mathrm{Si}) \ kT/q$	Dn	20.7 cm ² /s
$D_p = \mu_p (\mathrm{Si}) kT/q$	Dp	5.17 cm ² /s
τ_n	taun	~0.1 µs
τ_p	taup	~0.1 µs
$E_G(Si)$	Eg_Si	1.08 eV
χ_{Si}	X_Si	4.0 eV
χ_{SiO2}	X_SiO2	0.3 eV
$\chi_{\rm poly}$	X_poly	4.2 eV

The material parameters in the MOS transistor are:

INITIAL-VALUE CALCULATION

The rapid changes and large dynamics of the solution variables in the PDEs require a special technique to calculate the initial value. This model solves an extra electrostatics application mode separately to get the initial value for the other application modes. This extra application mode also solves Poisson's equation but with the difference that it replaces the carrier concentrations with the formulas

$$n = n_i e^{\frac{q \psi + \chi_{s_i} + \frac{E_g}{2}}{kT}}$$
$$p = n_i e^{\frac{q \psi + \chi_{s_i} + \frac{E_g}{2}}{kT}}$$

This step produces the exact solution for the full system when all applied voltages are zero. You perform the steps to get the initial condition with the solver scripting functionality in the **Solver Manager** dialog box.

Results and Discussion

The system's nonlinear behavior makes it necessary to use the parametric solver to reach the desired voltage for each contact. This example first raises the gate voltage to 0.8 V and then sweeps the drain voltage to 1 V. Including the first initial-value calculation, three solution steps are necessary, and by employing solver scripting you can execute all of them with one click on the **Solve** button. The result from the last step appears in Figure 12-9.



Figure 12-9: Drain current as a function of drain voltage for $V_{GS} = 0.8$ V. Note that R_{on} is 1/ (slope of the curve) at the point where $V_{DS} = 0$, which falls at the far left of the curve.

A surface plot is useful to view the shape of the solution variables. Figure 12-10 shows the electrostatic potential at the final bias condition. The large negative shift for the potential is due to the vacuum reference potential.



Figure 12-10: Surface plot of the electrostatic potential inside the MOSFET.

The conducting channel at the oxide-semiconductor interface becomes visible in a logarithmic surface plot of the electron concentration (Figure 12-11).



Figure 12-11: The channel close to the interface becomes visible in an electron-concentration plot. Notice the sharp drop in the channel concentration close to the drain contact, which is responsible for the saturation of the drain current.

Model Library path: COMSOL_Multiphysics/Semiconductor_Devices/ MOS_transistor

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator, select 2D from the Space dimension list.
- 2 Click the Multiphysics button.
- **3** Select the **COMSOL Multiphysics>Electromagnetics>Electrostatics** application mode.
- 4 Enter phi0 in the Dependent variables edit field, and enter init in the Application mode name edit field.

- 5 Click Add.
- 6 Enter phi in the Dependent variables edit field.
- 7 Click Add.
- 8 Select the COMSOL Multiphysics>Convection and Diffusion> Convection and Diffusion>Steady-state analysis application mode.
- **9** Enter nc in the **Dependent variables** edit field, and enter cde in the **Application mode name** edit field.
- IO Click Add.
- II Click the Application Mode Properties button, select Conservative from the Equation form list and On from the Weak constraints list. Click OK.
- 12 Enter pc in the Dependent variables edit field, and enter cdh in the Application mode name edit field.
- I3 Click Add.
- 14 Click the Application Mode Properties button, select Conservative from the Equation form list and On from the Weak constraints list. Click OK.
- I5 Click OK.

OPTIONS AND SETTINGS

- I From the **Options** menu, choose **Constants**.
- **2** In the **Constants** dialog box, define the following constants with names, expressions, and descriptions (the descriptions are optional):

NAME	EXPRESSION	DESCRIPTION
q	1.602e-19[C]	Elementary charge
то	300[K]	Lattice temperature
k	1.38e-23[J/K]	Boltzmann's constant
epsilonr_si	11.8	Rel. permittivity for Si
epsilonr_sio2	4.2	Rel. permittivity for SiO2
ni	1.5e10[1/cm^3]	Intrinsic carrier concentration for Si
mun_si	1000[cm^2/(V*s)]	Electron mobility for Si
mup_si	500[cm^2/(V*s)]	Hole mobility for Si
X_si	4.0[V]	Electron affinity for Si
X_sio2	0.3[V]	Electron affinity for SiO2
X_poly	4.5[V]	Work function for poly-Si

NAME	EXPRESSION	DESCRIPTION
Eg_si	1.08[V]	Band gap of Si
taun	0.1[um]	Electron carrier life time
taup	0.1[um]	Hole carrier life time
Vt	k*T0/q	Thermal voltage
x1	-0.1[um]	Help coordinate
x2	0.2[um]	Help coordinate
уј	0.1[um]	Junction depth
NDimpl	1e19[1/cm^3]	Peak concentration of implanted profile
NAsub	5e17[1/cm^3]	Substrate doping
ch	yj/sqrt(log(NDimpl/NAsub))	Characteristic length of implanted region
Vg	0.8[V]	Gate voltage
Vd	0[V]	Drain voltage

3 Click OK.

- 4 From the **Options** menu, choose **Functions**.
- 5 In the Functions dialog box, click the New button.
- 6 In the New Function dialog box, type flgauss in the Function name edit field. Click OK to create the new function.
- 7 Type x, y, x1, y1, ch in the Arguments edit field, and type the following in the Expression edit field: (exp(-((y-y1)/ch)^2)*(y<y1)+(y>=y1))* (exp(-((x-x1)/ch)^2)*(x>x1)+(x<=x1)).</p>
- 8 In the **Derivatives** area, click the **Manual** button and type diff(flgauss(x,y,x1,y1,ch),x),diff(flgauss(x,y,x1,y1,ch),y),0,0,0 in the corresponding edit field. The diff operator provides differentiation of the flgauss function with respect to x and y. The derivatives of flgauss with respect to x1, y1, and ch are all equal to 0.
- 9 Click OK.

GEOMETRY MODELING

I Draw rectangles with the properties according to the following table.

NAME	WIDTH	HEIGHT	CORNER	DESCRIPTION
RI	1e-6	5e - 7	(-4e-7,-5e-7)	Substrate

NAME	WIDTH	HEIGHT	CORNER	DESCRIPTION
R2	2.2e-7	5e-9	(-1e-8,0)	Gate oxide
R3	3.9e-7	5e-8	(-4e-7,0)	Air region

- **2** Click the **Zoom Extents** toolbar button.
- **3** Click the **2nd Degree Bézier Curve** button and draw an arc from (-1e-7, -1e-7), with control point (0, -1e-7), ending it at (0, 0).
- 4 Continue the arc with straight lines by clicking the Line button and then click the coordinate sequence (-4e-7, 0) and (-4e-7, -1e-7). Close to a solid by clicking the right mouse button.

Drawing a curve at coordinates with many decimals or using an expression can be difficult, and you might be forced to specify extra grid points. Here is an alternative approach. First draw a curve with a similar shape as the final curve, using the existing grid points. Then alter the properties of the curve segments with the **Object Properties** dialog box. Do the following to draw the small spacer regions on the sides of the poly-gate:

- **5** Click the **2nd Degree Bézier Curve** button and draw an arc from (-1e-7, 0), with control point (-1e-7, 1e-7), ending it at (0, 1e-7).
- 6 Continue the arc with straight lines by clicking the Line button and then click the coordinate sequence (0, 0) and (-1e-7, 0). Close to a solid by clicking the right mouse button.
- 7 From the Draw menu, choose Object Properties.
- 8 In the **Object Properties** dialog box, change the properties for each curve segment according the table below.

CURVE NUMBER	POINT NUMBER (ROW IN TABLE)	x	Y	WEIGHT
I	I	-5e-8	0	1
	2	-1e-8	0	1
2	I	-1e-8	0	1
	2	-1e-8	5e-8	1
3	I	-5e-8	0	1
	2	-5e-8	5e-8	1/sqrt(2)
	3	-1e-8	5e-8	1

9 Select the objects CO1, CO2, and R3, then click the **Mirror** button on the Draw toolbar.

10 In the Mirror dialog box, enter 1e-7 in the x edit field at the Point on line row. Leave all other fields at their default values and click OK. The geometry should look like the figure below.



PHYSICS SETTINGS

Expression Variables

- I From the **Options** menu, choose **Expressions**>Scalar **Expressions**.
- **2** In the **Scalar Expressions** dialog box, define the following variables with names and expressions, and descriptions (the **Description** field is optional):

NAME	EXPRESSION	DESCRIPTION
Dn	Vt*mun	Electron diffusivity
Dp	Vt*mup	Hole diffusivity
n0	ni*exp((phiO+X_si+ 0.5*Eg_si)/Vt)	Electron concentration in thermal equilibrium
p0	ni*exp(-(phiO+X_si+ 0.5*Eg_si)/Vt)	Hole concentration in thermal equilibrium
R_srh	(nc*pc-ni^2)/(taun*(pc+ni)+ taup*(nc+ni))	Shockley-Reed-Hall recombination term

NAME	EXPRESSION	DESCRIPTION
phi_init	<pre>Vt*(-log(p_init/ni)* (Ndoping<0)+log(n_init/ni)* (Ndoping>=0))-X_si -0.5*Eg_si</pre>	Initial guess for phiO
n_init	<pre>(abs(Ndoping)/2+ sqrt(Ndoping^2/4+ni^2))* (Ndoping>=0)+ ni^2/(abs(Ndoping)/2+ sqrt(Ndoping^2/4+ni^2))* (Ndoping<0)</pre>	Electron concentration at thermal equilibrium and charge neutrality
p_init	<pre>(abs(Ndoping)/2+ sqrt(Ndoping^2/4+ni^2))* (Ndoping<0)+ ni^2/(abs(Ndoping)/2+ sqrt(Ndoping^2/4+ni^2))* (Ndoping>=0)</pre>	Hole concentration at thermal equilibrium and charge neutrality
Ndoping	NDimpl*flgauss(x,y,x1,0,ch) -NAsub+NDimpl* flgauss(x2-x,y,x1,0,ch)	Doping concentration

3 Click OK.

Subdomain Variables

I From the **Options** menu, choose **Expressions>Subdomain Expressions**.

2 In the Subdomain Expressions dialog box, define the following variables:

VARIABLE	SUBDOMAINS 1, 2, 6	SUBDOMAINS 4, 5, 7	ALL OTHER SUBDOMAINS
epsilonr	epsilonr_si	epsilonr_sio2	1
mun	mun_si		
mup	mup_si		

3 Click OK.

Boundary Variables

- I From the Options menu, choose Expressions>Boundary Expressions.
- **2** In the **Boundary Expressions** dialog box, define the following variables with names and expressions:

VARIABLE	BOUNDARIES 6, 19	ALL OTHER BOUNDARIES
Jn	-q*lm3	
Jp	q*lm4	

3 Click OK.

Integration Coupling Variables

The drain current Id is defined as an integral of the normal current density along the drain contact. This can be implemented by using an integration coupling variable:

- I From the Options menu, select Integration Coupling Variables>Boundary Variables.
- 2 In the Boundary Integration Variables dialog box, select Boundary 19 and then type Id in the Name column, and type Jn+Jp in the Expression column. Click OK.

Subdomain Settings

- I On the Multiphysics menu, select the Electrostatics (init) application mode.
- 2 From the Physics menu, select Subdomain settings.
- 3 In the Subdomain Settings dialog box, enter the following settings:

SETTINGS	SUBDOMAINS 1, 2, 6	ALL OTHER SUBDOMAINS
ϵ_r (isotropic)	epsilonr	epsilonr
ρ	q*(p0-n0+Ndoping)	0

4 Click the **Init** tab, and enter the following initial value:

SETTING	ALL SUBDOMAINS	
phi0(t ₀)	phi_init	

5 Click OK.

6 On the Multiphysics menu, select the Electrostatics (es) application mode.

7 Open the Subdomain Settings dialog box and enter the following settings:

SETTINGS	SUBDOMAINS 1, 2, 6	ALL OTHER SUBDOMAIN
ϵ_r (isotropic)	epsilonr	epsilonr
ρ	q*(pc-nc+Ndoping)	0

8 Click the Init tab, and enter the following initial value:

SETTING	ALL SUBDOMAINS	
phi(t ₀)	phi0	

9 Click OK.

10 On the **Multiphysics** menu, select the **Convection and Diffusion (cde)** application mode.

II Open the **Subdomain Settings** dialog box, select Subdomains 3, 4, 5, 7, and 8, and clear the **Active in this subdomain** check box. Then enter the following settings for the other subdomains:

SETTINGS	SUBDOMAINS 1, 2, 6	
D (isotropic)	Dn	
R	-R_srh	
u	mun*phix	
v	mun*phiy	

12 Click the **Init** tab and enter the following initial value:

SETTING	SUBDOMAINS 1, 2, 6	
nc(t ₀)	n0	

I3 Click OK.

- **14** On the **Multiphysics** menu, select the **Convection and Diffusion (cdh)** application mode.
- **IS** Open the **Subdomain Settings** dialog box, select Subdomains 3, 4, 5, 7, and 8, and clear the **Active in this subdomain** check box. Then enter the following settings for the other subdomains

SETTINGS	SUBDOMAINS 1, 2, 6		
D (isotropic)	Dp		
R	-R_srh		
u	-mup*phix		
v	-mup*phiy		

I6 Click the **Init** tab, and enter the following initial value:

SETTING	SUBDOMAIN 1, 2, 6	
pc(t ₀)	p0	

I7 Click OK.

Boundary Conditions

- I On the Multiphysics menu, select the Electrostatics (init) application mode.
- 2 From the Physics menu, select Boundary Settings. Select the Interior boundaries check box.

SETTINGS	BOUNDARIES 2, 6, 19	BOUNDARIES 11, 12, 17	BOUNDARIES 1, 3, 5, 7, 18, 21–23
Boundary condition	Electric potential	Electric potential	Zero charge/Symmetry
V ₀	phi_init	Vg-X_poly	

3 In the **Boundary Settings** dialog box, enter the following settings:

4 Click OK.

- 5 On the Multiphysics menu, select the Electrostatics (es) application mode.
- 6 Open the **Boundary Settings** dialog box, select the **Interior boundaries** check box, and enter the following settings:

SETTINGS	BOUNDARIES 2, 6	BOUNDARY 19	BOUNDARIES	BOUNDARIES 1, 3, 5, 7, 18, 21–23
Boundary condition	Electric potential	Electric potential	Electric potential	Zero charge/ Symmetry
V ₀	phi_init	phi_init+Vd	Vg-X_poly	

- 7 Click OK.
- 8 On the Multiphysics menu, select the Convection and Diffusion (cde) application mode.
- 9 Open the **Boundary Settings** dialog box and enter the following settings:

SETTINGS	BOUNDARIES 2, 6, 19	ALL OTHER BOUNDARIES
Boundary condition	Concentration	Insulation/Symmetry
nc ₀	n_init	

IO Click OK.

- II On the Multiphysics menu, select the Convection and Diffusion (cdh) application mode.
- 12 Open the Boundary Settings dialog box and enter the following settings:

SETTINGS	BOUNDARIES 2, 6, 19	ALL OTHER BOUNDARIES
Boundary condition	Concentration	Insulation/Symmetry
pc_0	p_init	

I3 Click OK.

MESH GENERATION

I From the Mesh menu, choose Free Mesh Parameters.

- **2** Click the **Custom mesh size** button and enter 0.2 in the **Mesh curvature factor** edit field. This gives a slightly higher resolution of the curvature than the default setting.
- **3** Click the **Subdomain** tab.
- 4 Select Subdomains 2 and 6 and type 1e-8 in the Maximum element size edit field.
- **5** Click the **Boundary** tab.
- 6 Select Boundaries 4, 20, 24, and 26 and type 5e-9 in the Maximum element size edit field.
- 7 Select Boundary 13 and type 2e-9 in the Maximum element size edit field.
- 8 Click the **Remesh** button and then click **OK**.

COMPUTING THE SOLUTION

- I From the Solve menu, choose Solver Manager.
- 2 In the Solver Manager dialog box, click on the Script tab.
- **3** Click the **Solve using a script** check box, and enter the following script in the large text area.

```
init = asseminit(fem);
fem.sol = femstatic(fem, ...
    'init',init, ...
    'solcomp',{'phi0'}, ...
    'outcomp',{'lm3','phi','phi0','pc','nc','lm4'});
fem0 = fem;
init = asseminit(fem, 'u',fem0.sol);
fem.sol = femstatic(fem, ...
    'init',init, ...
    'u',fem0.sol, ...
    'solcomp',{'lm3','phi','pc','nc','lm4'}, ...
    'outcomp',{'lm3','phi','phi0','pc','lm4','nc'}, ...
    'pname','Vd', ...
    'plist',[0:0.02:0.2 0.25:0.05:1]);
```
femO = fem;

Solver Manager
Initial Value Solve For Output Script
✓ Solve using a script
<pre>init = asseminit(fem); fem.sol=femstatic(fem,</pre>
<pre>init = asseminit(fem, 'u',fem0.sol); fem.sol=femstatic(fem,</pre>
۲ III ا
Automatically add commands when solving Add Current Solver Settings
Solve OK Cancel Apply Help

4 Click **OK** and then click the **Solve** button to start the simulation. Note that the script overrides all other settings in the **Solver Parameters** and **Solver Manager** dialog boxes.

POSTPROCESSING AND VISUALIZATION

- I Open the **Plot Parameters** dialog box.
- **2** Click the **Surface** tab.
- 3 On the Surface page, enter phi in the Expression edit fields on both the Surface Data tab and the Height Data tab. Remember to select the Height data check box on the Height Data tab.

4 Click **OK** to get the plot in the figure below.



- **5** To plot the drain characteristics open the **Domain Plot Parameters** dialog box from the **Postprocessing** menu.
- 6 Click the **Point plot** button to select that plot type and make sure that all parameter values are selected in the **Solutions to use** list.
- 7 On the Point tab, type Id in the Expression edit field in the y-axis data area.



8 Select vertex 1 and click **OK**. In a separate window you should see the plot below.

13

Structural Mechanics Models

This section contains a modal analysis of a crankshaft based on a NASTRAN mesh; a 3D model of a feeder clamp; a parametric study of the stresses in a pulley; a model of a communication mast's diagonal mounting; and a model of a Formula One car's front wing. The last two models, both in 3D, highlight the use of assemblies and interactive meshing. The *COMSOL Multiphysics Modeling Guide* includes two benchmark models using 2D stress analysis. A comprehensive set of solved examples is available in the Model Library that comes with the Structural Mechanics Module.

Eigenvalue Analysis of a Crankshaft

This model describes a modal analysis of a crankshaft. The pistons' reciprocating movement is transferred to the crankshaft through connecting rods by means of crankshaft throws. The forces, torques, and bending moments, which are highly variable both in time and space, subject the crankshaft to very high and complex loading. The crankshaft design must therefore incorporate careful and precise calculations of the vibrational characteristics.

Model Definition

GEOMETRY

The geometry comes from a NASTRAN mesh, which you import into the COMSOL Multiphysics user interface.

MATERIAL PROPERTIES AND BOUNDARY CONDITIONS

The crankshaft geometry is in millimeters, and a suitable unit system for geometries given in millimeters is the MPa system. The crankshaft is made of solid steel, and you give the material properties in the MPa unit system; that is, express the Young's modulus in MPa and the density in t/mm³.

The boundary conditions are as follows:

- The crankshaft's main bearing surfaces are constrained from moving in the normal directions; that is, the crankshaft is allowed to rotate and slide at the bearing surfaces.
- The axial movement is constrained at the rear axial bearing surface.
- The crankshaft is fixed at the rear surface where the flywheel is mounted.

Results

The analysis provides the 20 first eigenfrequencies, of which several of them show the same modal shapes. This first eigenmode has an eigenfrequency at approximately 5778 Hz and it has a torsional characteristic. There are five additional eigenfrequencies in rather close vicinity to this frequency. Each of them also shows similar torsional characteristics, which suggests that they are all in fact showing the first torsional mode.



Figure 13-1: The first torsional mode in the crankshaft.

The next distinctly shaped eigenmode has an eigenfrequency at 8132 Hz. It can be characterized as a bending mode in the crankshaft throws. Here there are also five additional eigenfrequencies in rather close vicinity to this frequency. Each of these frequencies shows similar bending shapes as in the previously mentioned eigenmode at 8132 Hz. This suggests that they are all in fact showing the first bending mode in the crankshaft throws.



Figure 13-2: The bending mode of the crankshaft throws.

The third distinctly shaped eigenmode has an eigenfrequency at 8891 Hz. This modal shape can be characterized as a twisting mode in the crankshaft throws. There are, in the same manner as earlier, five additional eigenfrequencies in rather close vicinity to this frequency. Each of these frequencies shows similar twisting shapes.



Figure 13-3: The twisting mode of the crankshaft throws.

Model Library path: COMSOL_Multiphysics/Structural_Mechanics/ crankshaft

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Open the Model Navigator, and in the Space dimension list select 3D.
- **2** In the list of application modes select

COMSOL Multiphysics>Structural Mechanics>Solid, Stress-Strain>Eigenfrequency analysis.

3 Click OK.

GEOMETRY MODELING

- I From the File menu choose Import and then Mesh from File.
- 2 Browse to the NASTRAN mesh file located at models/COMSOL_Multiphysics/ Structural_Mechanics/crankshaft.nas in the COMSOL installation directory.
- **3** Click the **Import** button.



PHYSICS SETTINGS

Model Settings

- I From the Physics menu choose Model Settings.
- 2 In the Base unit system list select MPa.
- 3 Click OK.

Boundary Conditions

I From the Physics menu choose Boundary Settings.



2 In the Boundary selection list choose 27, 55, 65, and 68–74.

3 Click the General notation, Hu=R button, then click the Edit button associated with the H matrix. In the matrix position (1, 1) enter nx, in the (1, 2) position enter ny, and in the (1, 3) position enter nz. Click OK.

These settings constrain the normal direction movements of the main bearing surfaces.

- 4 In the **Boundary selection** list choose 26.
- 5 Click the Standard notation button, then select the R_x, R_y, and R_z check boxes to fix this surface.
- 6 Click OK.

Subdomain Settings

The crankshaft geometry is in millimeters. This means that you must enter the material properties in the MPa unit system to avoid having to scale the geometry; that is, you specify the Young's modulus in MPa and the density in t/mm³.

I From the Physics menu choose Subdomain Settings.

2 Enter the following material properties:

SUBDOMAIN	I
E	2.0e5
ν	0.33
ρ	7850e-12

- **3** Click the **Element** tab.
- **4** Select **Lagrange Linear** in the **Predefined elements** list. The model uses linear elements to conform with the NASTRAN mesh.
- 5 Click OK.

COMPUTING THE SOLUTION

- I From the Solve menu choose Solver Parameters.
- 2 In the General page go to the Desired number of eigenfrequencies edit field and enter20. Click OK.
- 3 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

COMSOL Multiphysics plots the first eigenmode using a slice plot. To change the plot to show the deformed shape and other eigenmodes, follow these steps:

- I From the Postprocessing menu choose Plot Parameters.
- 2 In the **Plot type** area on the **General** page, clear the **Slice** check box, then select the **Boundary** and **Deformed shape** check boxes.
- 3 Select the second eigenfrequency in the Eigenfrequency list and click Apply to plot it.
- **4** Select the last eigenfrequency in the **Eigenfrequency** list and click **OK** to plot the twentieth eigenmode and close the **Plot Parameters** dialog box.

Deformation of a Feeder Clamp

Introduction

This example, from the field of structural mechanics, analyzes the deformation of a feeder clamp under stress. The clamp secures a feeder that carries high-frequency electromagnetic fields, and it's important that it remains as straight as possible.

This example analyzes deformations in the clamp with two key questions in mind:

- How much does the force from the feeder incline the clamp? The inclination must be less than 1 degree.
- Does a prestressed screw of a certain type have enough strength to deform the clamp so that it adequately anchors the feeder? The gap must shrink by at least 0.5 mm

Model Definition

Forces on the clamp include those from the feeder as well as those on the sleeve arising from the clamping screw. In this analysis, the feeder clamp is attached to a wall using only one of its two mounting holes, so that you can better analyze how it performs when poorly secured.



LOADING DATA

Assume that an installation technician fastens the feeder into the clamp using a standard metric M3 screw of Class 8.8 (where the first digit stands for a breaking load of 800 N/mm², and the second digit indicates a yield strength of 80% of the breaking strength). Prestressing the screw to the yield limit results in a screw force of 4500 N. This model tests if a screw force of 80% of this is adequate. A 7-mm washer distributes the prestressed load evenly onto both sides of the sleeve. The maximum load from the feeder onto the clamp is 1000 N, and is applied evenly throughout the inside of the clamp.

Due to symmetry in both loading and the geometry, you can perform a complete model analysis while looking at only one half of the geometry. For illustrative purposes, though, this example models the entire geometry.

Results and Discussion



The z-deformation for the whole clamp is depicted in Figure 13-4.

Figure 13-4: z-deformation.

The maximum inclination about the *z*-axis can be derived from the difference between the maximum and the minimum deformation in the *y* direction. The maximum and minimum values can be found from the colormap at the right in Figure 13-5. It can be shown that the sine of the maximum inclination is equal to the difference between the maximum and the minimum *y*-deformation, divided by the width of the clamp in the *z* direction. For small angles, a, this becomes

$$\alpha = \frac{0.094 - (-0.014)}{20} \cdot \frac{180}{\pi}$$

which gives an inclination of 0.31 degrees which is less than the maximum allowed angle of 1 degree.



Figure 13-5: Maximum and minimum y-displacement at the center of the clamp cylinder.

To see the maximum displacement in the x direction, you can plot abs(u) > 0.25 mm. Figure 13-6 shows that the deformation in the feeder clamp gap is larger than 0.25 mm on both sides, which is the minimum requirement for being able to fasten the feeder adequately. The boundary conditions in this model are valid only as long as the installation technician does not squeeze the gap in the clamping sleeve completely shut. However, if the gap is squeezed shut, you can be sure that the requirements are fulfilled.



Figure 13-6: Clamp deformation due to the screw force.

Model Library path: COMSOL_Multiphysics/Structural_Mechanics/ feeder_clamp

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Start COMSOL Multiphysics. This invokes the Model Navigator.
- 2 On the New page, select 3D from the Space dimension list.
- **3** Double-click the **COMSOL Multiphysics>Structural Mechanics** folder and select **Solid**, **Stress-Strain>Static analysis**.
- 4 Click OK.

Notice that this gives you second-order elements, Lagrange - Quadratic, by default.



OPTIONS AND SETTINGS

- I Double-click **AXIS** on the status bar to turn off the 3D axis.
- 2 Choose Model Settings from the Physics menu and select MPa from the Base unit system list. Click OK.
- 3 Choose Constants from the Options menu.
- **4** Enter the following constant names, expressions, and descriptions (optional):

NAME OF CONSTANT	EXPRESSION	DESCRIPTION
Ffeeder	1000	Feeder force
Afeeder	pi*20*20	Feeder area
Fscrew	3600	Screw force
D	7	Outer diameter
d	3.75	Inner diameter
Awasher	pi/4*(D^2-d^2)	Washer area

To add a constant, enter its name in the **Name** edit field and its corresponding constant value or defining equation in the **Expression** edit field. Adding a description

in the **Description** edit field is optional. After you finish entering constants, click **Apply** to evaluate all the expressions.

Name	Expression	Value	Description	
Ffeeder	1000	1000	Feeder force	
Afeeder	pi*20*20	1256.637061	Feeder area	
Fscrew	3600	3600	Screw force	
D	7	7	Outer diameter	
d	3.75	3.75	Inner diameter	
Awasher	pi/4*(D^2-d^2)	27.439848	Washer area	
				=

5 Click OK.

GEOMETRY MODELING

Creating the First Work-Plane Geometry

- I Choose **Work-Plane Settings** from the **Draw** menu and click **OK** in the dialog box to enter the default work plane, which is the *xy*-plane.
- 2 Choose Axes/Grid Settings from the Options menu to open the corresponding dialog box.
- **3** On the **Axis** page, type -40, 70, -5, and 65 in the **x** min, **x** max, **y** min, and **y** max edit fields, respectively. Click **Apply** to update the axis settings.

Axes/Grid Settings Axis Grid	X
Axis equal	7 limits
x min: -40 x max: 70 y min: -5 y max: 65	Z min: -1 z max: 1
40	Cancel Apply Help

- 4 Click the Grid tab. Clear the Auto check box, and type 5 in both the x spacing and y spacing edit fields. Add extra grid lines at x = 14.25 and x = 15.75 by entering these values, with a space separating them, in the Extra x edit field. Click OK.
- 5 Using the left mouse button, draw a rectangle with corners at (5, 5) and (25, 60). Use the Rectangle/Square button on the Draw toolbar.

- 6 Draw a circle with a radius of 15 by selecting the Ellipse/Circle (Centered) toolbar button. Using the right mouse button, place the center of the circle at (15, 35) and drag the mouse until you have the correct radius. During dragging, you can monitor the current value of the radius in the coordinate field to the left in the Status bar.
- 7 Press Ctrl+A to select both geometry objects.
- 8 Click the **Difference** button on the Draw toolbar to remove the circle from the rectangle.



- 9 Draw another circle with the same radius at the same position as the deleted one. Then draw yet another circle centered at (15, 35), but give this one a radius of 10. Select the Ellipse/Circle (Centered) button on the Draw toolbar and then click-and-drag using the right mouse button to create a circle.
- **10** Draw a rectangle with corners at (14.25, 40) and (15.75, 60) by choosing the **Rectangle/Square** button and using the extra grid lines.
- II Click the Create Composite Object button on the Draw toolbar. In the Set formula edit field of the dialog box that opens enter the Boolean expression (C01+C1) (C2+R1). Click OK to create the composite solid object CO2.

Create Composite Object			X
Object type <u>Solids</u> Curves Points		Shortcuts Union Intersection Select All	OK Cancel Apply Help
Object selection: CO1 C1 C2 R1	*	Set formula: (CO1+C1)-(C2+R1) Veep interior bour Repair Repair Verpair 10erance:	ndaries

- **12** If you make a mistake, choose **Undo** from the **Edit** menu to revert to the previous state.
- **13** Split the created composite solid object into its constituents. To do so, click the **Split Object** button on the Draw toolbar.
- 14 Click the **Rectangle/Square** toolbar button and use the left mouse button to make a rectangle with corners at (0, 0) and (30, 5).



Extruding the 2D Geometry Objects into 3D Solid Objects

- I To extrude the 2D rectangle into a 3D object, choose **Extrude** from the **Draw** menu. This selection opens the **Extrude** dialog box. Make sure that the last created rectangle is selected and enter 40 in the **Distance** edit field before clicking **OK**.
- **2** The resulting 3D geometry object appears in a 3D view. Click **Zoom Extents** in the main toolbar to view the entire block.



3 Click the Move button in the Draw toolbar; in the resulting dialog box first make sure the 3D block is selected. Then enter a z-displacement of -10 and click **OK**.

Move	X
Displacement x: y: 0 z: -10	OK Cancel

- 4 Enter the work plane again by selecting I Geom2 (2D) from the Draw menu. Click the Projection of All 3D Geometries button at the lower part of the Draw toolbar.
- **5** While working in the plane, open the **Extrude** dialog box from the **Draw** menu and select the objects CO1, CO3, and CO5 that came from the first rectangle you created. Enter an extruding distance of 10. Click **OK** to extrude the 2D geometry and enter the 3D view.

- 6 Make sure that the recently created 3D objects are selected. Click the Move button on the Draw toolbar. In the resulting dialog box enter 5 in the z edit field and click OK.
- 7 Re-enter the work plane by selecting I Geom2 (2D) from the Draw menu.
- 8 Select the circular-shaped solid object, object CO4, and open the **Extrude** dialog box. Enter a distance of 20 and click **OK**. Click **Zoom Extents** in the Main toolbar to see the entire resulting geometry. Click and drag in the figure to rotate the geometry to get the desired view.



Creating a Second Work Plane

I It is now necessary to add a work plane, so choose **Work-Plane Settings** from the **Draw** menu. In the resulting dialog box, click the **Add** button. The software creates a work

plane and automatically assigns it the name Geom3. Select the **Quick** page. Click the **y-z** button and specify an offset of 5 in the **x** edit field.

Work-Plane Settings	X
Quick Face Parallel Edge Angle Vertices Advanced Plane	Cancel Apply Help Preview
Workplane (2D geometry): Geom3 🗸 Add	

- 2 Click OK to enter the work plane environment and then click the Zoom Extents button in the work plane. The 3D projection of the geometry onto the work plane is visible if any of the Projection of All 3D Geometries or Project Work-Plane Intersection buttons are selected at the bottom of the Draw toolbar. You can snap to the projection when drawing.
- 3 Choose Axes/Grid Settings from the Options menu. In the resulting dialog box, go to the Grid page, uncheck the Auto check box and enter a grid spacing of 5 in both the x spacing and y spacing fields. Add extra grid lines at x = 56.875 and x = 58.5 by entering these values, separated with a space, in the Extra x field. Click OK to close the dialog box.
- **4** Draw a circle with its center at (55, 10) by choosing **Ellipse/Circle (Centered)** and clicking the right mouse button. Drag the cursor from the center to the right and release the button over one of the extra grid lines. Draw another circle centered at the same position, but this time release the mouse button over the other extra grid line. Now you have two circles with radii of 1.875 and 3.5.
- **5** Press Ctrl+A to select both circles. Click the **Difference** button on the Draw toolbar to subtract the smaller circle from the larger one. The resulting object is given the name CO1.
- 6 Redraw the smaller of the two circles. The circle is given the name C1 by the software. Additionally draw a new circle with a radius of 5 with the same center coordinates. It will be named C2. Click the **Rectangle/Square** button and use the left mouse button to draw a rectangle with corners at (55, 5) and (60, 15).

7 Open the **Create Composite Object** dialog box by Shift-clicking the associated button on the Draw toolbar. In the **Set formula** edit field type R1-C2+C1. Click **OK** to create the composite solid object CO2.



Figure 13-7: Extruding the 2D Geometry Object and Creating 3D Solid Object

- I Choose **Extrude** from the **Draw** menu. In the resulting dialog box make sure the last created object, CO2, is selected. Then enter 9.25 in the **Distance** edit field and click **OK**.
- 2 In 3D view, click the **Copy** button in the main toolbar and then click the **Paste** button. In the **Displacements** area, enter an *x*-displacement of 10.75 in the **x** edit field, and click **OK**.
- 3 Return to the work plane by clicking on the **Geom3** tab above the drawing area.
- 4 To define the area of the washer that distributes pressure from the screw, choose Embed from the Draw menu. In the resulting dialog box, select CO1 and click OK.
- 5 In 3D view, click the Copy button in the main toolbar and then click the Paste button. In the Displacements area, enter an *x*-displacement of 20 in the x edit field, and click OK.

6 To round off the top of the clamp and create a hole for the clamping screw, subtract solid object EXT6 and EXT7 from objects EXT3 and EXT4 using the Create Composite Object dialog box.

To do so, click the **Create Composite Object** toolbar button to open the dialog box and select the **Solids** option button in the **Object type** frame.

- △▲ === @ @ @ @ # # W to to to 0 @ ? 0 🖬 💟 Geom2 Geom2 ti ti ti 8 0 10 ā 0 医屈围区 30 H i
- 7 Type (EXT3+EXT4)-EXT6-EXT7 in the Set formula edit field. Click OK.

8 You can add lighting to the plot by clicking the **Headlight** button in the Camera toolbar.

Creating a Third Work Plane and the Final 3D Solid Model

- I Press Ctrl+A to select all geometry objects, then choose Work-Plane Settings from the Draw menu.
- 2 In the dialog box, click the Add button to create a new work plane, which automatically gets the name Geom4. On the Face Parallel page select the folder/ solid object EXT1 and the face number 2 from the list. Click the Downward normal button. Click OK to enter the new work plane.
- 3 Select Axes/Grid Settings from the Options menu. In the resulting dialog box, click the Grid tab, clear the Auto check box, and enter a grid spacing of 2.5 in both the x spacing and y spacing fields. Add extra grid lines at x = 9.5 and x = 20.5 by entering

these values, separated by a space, in the **Extra x** edit field. Click **OK** to close the dialog box.

- **4** Draw a rectangle with corners at (9.5, -10) and (20.5, -5) using the left mouse button and the **Rectangle/Square** toolbar button.
- 5 Click the Ellipse/Circle (Centered) button, and use the right mouse button to draw a circle centered at (9.5, -7.5) with a radius of 2.5. Draw a similar circle centered at (20.5, -7.5).
- 6 Press Ctrl+A to select all objects. Click the **Copy** button in the main toolbar, then click the **Paste** button. In the **Displacements** area, enter a *y*-displacement of -25 in the **y** edit field, and click **OK**.



7 Press Ctrl+A to select all objects and click the Union button in the Draw toolbar.

- 8 Choose Extrude from the Draw menu. In the dialog box, type 5 in the Distance edit field. Click OK to enter 3D view.
- **9** In 3D view, the recently extruded objects are already selected. Holding down the Shift key, click on the block you created first, EXT1, using the left mouse button to add it to the selection. Then click the **Difference** toolbar button to create the mounting holes in the clamp.

Boundary Conditions

The boundary conditions consist of loads and constraints. The following tables list the loads (top) and constraints (bottom):

	BOU	NDARY 25	BOUNDARY 68			IBOUNDARIES 7, 18, 45, 47		
Page	Loa	d	Load		Load			
	F_x	Fscrew/Awasher	F_x	-Fscrew/Awasher	F_x	0		
	Fy	0	Fy	0	Fy	0		
	Fz	0	Fz	0	Fz	-Ffeeder/Afeeder		

	BOUNDARIES 32, 33, 36, 37, 60, 61 Constraint		
Page			
	R _x	0	
	Ry	0	
	R _z	0	

The constraint means that the feeder clamp is fixed (no displacements) at the mount. All other boundaries are free.

In relatively complicated geometries like this it is often convenient to arrange the different boundaries in groups. This allows you to set properties for a group of domains at once. To simplify the specification of the boundary conditions, create five groups:

- clamp, containing the four boundaries which constitute the inside surface of the large cylinder. These are the boundaries where the feeder force is applied.
- mount, containing the six boundaries where the feeder clamp is attached.
- washer top, containing the top boundary where the screw force is applied.
- washer bottom, containing the bottom boundary, where the screw force is applied.
- free, containing the rest of the boundaries, which are free of loads and constraints.

The following steps describe, in addition to the boundary settings, how to create the boundary domain groups.

- I Open the **Boundary Settings** dialog box by selecting **Boundary Settings** from the **Physics** menu. This dialog box enables you to select boundaries and enter expressions for boundary conditions.
- 2 To create the boundary groups, click the **Groups** tab. Select the existing group (unnamed1), enter the name free in the Name edit field, and press Enter.

3 Click the New button and enter the name clamp, and press Enter. Similarly, add the remaining groups mount, washer top, and washer bottom.

GROUP	WASHER TOP		WASHER BOTTOM		CLAMP	
Page	Loa	d	Load		Loa	d
	F _×	Fscrew/ Awasher	F _x	-Fscrew/ Awasher	F _×	0
	Fy	0	Fy	0	Fy	0
	Fz	0	Fz	0	Fz	-Ffeeder /Afeeder

Now enter the boundary conditions for the groups:

GROUP	MOUNT		
Page	Constraint		
	R _×	0	
	Ry	0	
	R _z	0	

The **free** group should have the default settings, so make sure that all loads are zero for this group and that none of the displacement check boxes are selected on the **Constraint** tab.

- I Click the **Boundaries** tab and select Boundary 25, then choose **washer top** from the **Group** list.
- 2 Select Boundary 68, then choose washer bottom from the Group list.
- 3 Select Boundaries 17, 18, 45, and 47, then choose clamp from the Group list.
- **4** Select Boundaries 32, 33, 36, 37, 60, and 61, then choose **mount** from the **Group** drop down list.
- 5 Make sure the remaining boundaries belong to the free group.
- 6 Click **OK** to close the dialog box.

Subdomain Settings

The domain parameters for this model are:

SETTINGS	SUBDOMAINS I-5
E	2.06e5
ν	0.3

This example models the geometry in millimeters, which means that for consistency you must specify Young's modulus, E, in N/mm².

- I Open the Subdomain Settings dialog box by selecting Subdomain Settings from the Physics menu
- 2 Select all subdomains by pressing Ctrl+A.
- 3 Click the Material tab.
- **4** Ignore the value for the density, a parameter that has no effect on the solution in this static simulation.
- 5 Type 2.06e5 in the E edit field.
- **6** Type 0.3 in the edit field for Poisson's ratio, v.
- 7 Click OK.

Subdomain Settings - Solid, S	tress-Strain (sl	d)			X	
Subdomains Groups	Material Cor	nstraint Load Damping Init Ele	ement	Color		
Subdomain selection	Material settings					
1 ^	Library mat	erial: 🗾 🗸 Load				
3	Quantity	Value/Expression	Unit	Description		
4	E	2.06e5	MPa	Young's modulus		
5	v	0.3		Poisson's ratio		
	ρ	7.85e-9	t/mm ³	Density		
-						
Group: 🚽						
Select by group						
Active in this domain						
		OK Cancel		Apply Hel	p	

MESH GENERATION

Change the default mesh parameters to distribute the mesh elements in a more reasonable way. The maximal stresses in this model will be found in the sleeve, on the side facing the wall. Therefore, it makes sense to require a finer mesh in the sleeve than in the rest of the geometry. To accomplish this, do the following:

- I Choose Free Mesh Parameters from the Mesh menu.
- **2** Select **Coarser** from the **Predefined mesh sizes** list, which saves memory and computation time.
- **3** To obtain a higher mesh resolution in the sleeve (Subdomain 2), click the **Subdomain** tab, select Subdomain 2, and type **3** in the **Maximum element size** edit field. This imposes an upper bound of **3** mm on the element edge lengths in Subdomain 2.

- 4 Click the **Remesh** button.
- 5 Click OK.

COMPUTING THE SOLUTION

To start computing the solution, click the **Solve** toolbar button. COMSOL Multiphysics displays the results automatically as a slice plot of the von Mises stress.

POSTPROCESSING AND VISUALIZATION

- Choose Plot Parameters from the Postprocessing menu to open the Plot Parameters dialog box. On the General page, clear the Slice check box and then select the Boundary and Deformed shape check boxes to add the corresponding plot types. The Geometry edges plot is on by default.
- **2** To plot the deformation in the *z* direction, select *z*-displacement in the **Predefined quantities** field on the **Boundary** page.
- 3 Click Apply to display a plot using the new settings (see Figure 13-4).
- **4** To change the view of the solution plot, work with the buttons in the Camera toolbar.
- 5 To compute the maximum inclination in the *z* direction, you can derive from the difference between the maximum and the minimum deformation in the *y* direction. Select **y-displacement** from the **Predefined quantities** list on the **Boundary** page in the **Plot Parameters** dialog box.
- 6 On the General page, select the Element selection check box and type x==5 & y==35 in the Logical expression for inclusion edit field. Select At least one from the Element nodes to fulfill expression list and click OK. This plots the deformation on the inside of the feeder clamp. Click the Go to YZ View button to get a better view (see Figure 13-5).
- 7 To see the maximum displacement in the x direction, clear the Element selection check box on the General page in the Plot Parameters dialog box and type abs(u)>0.25 in the Expression edit field on the Boundary page. Then click OK.

To get a better view of the clamping sleeve, zoom in on the top of the clamp; click the **Go to XY View** button and use the **Zoom Window** button in the Main toolbar (see Figure 13-6).

Stresses in a Pulley

Introduction

This model contains a study of the stress distribution in a driving pulley. The analysis shows the stresses as functions of the pulley's angular velocity.

Model Definition

Figure 13-8 shows the pulley under study (to the right) and the external forces applied due to the driving belt.



Figure 13-8: Pulley and driving belt with the external forces F_1 and F_2 .

Here, F_1 and F_2 are the loads in the load side and in the slack side of the belt, respectively. The relationship between these forces is given by Reynold's equation (or Eytelwein's equation, as it is referred to in the German literature):

$$\frac{F_1}{F_2} = e^{\mu \cdot \beta}$$

where μ is the coefficient of friction and β is the contact angle between the belt and the pulley. This equation is valid if a condition of impending slippage between the belt and the pulley prevails.

It is also necessary to state that the peripheral force (the force that transmits the power) is

$$F_u = F_1 - F_2 = \frac{M}{r}$$

It is then possible to define F_2 as

$$F_2 = \frac{F_u}{(e^{\mu \cdot \beta} - 1)}$$

Now that you know the force in the slack side of the belt, you can derive the loads on the boundary of the pulley. In mechanics and the theory for flexible wires the following equilibrium equations appear:

$$F_n = \frac{S}{r}$$
$$F_t = \mu \cdot F_n$$

where F_n is the normal component and F_t the tangential component of the external forces, S is the tension force, and r is the radius.

If you apply these equations to this model, the loads on the boundary of the pulley become

$$F_n(\theta) = \frac{F_2}{r} \cdot e^{\mu \cdot \theta}$$

and

$$F_t(\theta) = \mu \cdot F_n(\theta)$$

where θ is the angle for which the forces are calculated (see Figure 13-8).

Due to the rotation of the pulley, inertia loads are generated. These loads can be calculated as

$$F_r = r \cdot \omega^2 \cdot \rho$$

where *r* is the radius, ω is the rotation speed, and ρ is the density.

The pulley is fixed at its inner diameter and the inertia loads are active in the entire geometry.

A parametric analysis shows how the rotational speed affects the stress distribution in the pulley. Because the power at the pulley shaft remains constant, the torque (defined as the ratio of the power by the rotational speed) decreases with increased speed. This means that with increased rotational speed, the inertial load increases while the driving-belt force decreases. The following plots show the von Mises stress distribution inside the pulley for different rotational speeds in rpm (revolutions per minute).

As is evident from the plots, the stress distribution changes as the rotational speed increases.



Figure 13-9: von Mises stress distribution at n = 1000 rpm.



Figure 13-10: von Mises stress distribution at n = 5000 rpm.



Figure 13-11: von Mises stress distribution at n = 9000 rpm.

At the point (0.0179, 0.0535), the von Mises stress is maximal for the first rotational speed (n = 1000 rpm). The plot in Figure 13-12 shows how the rotational speed affects the von Mises stress at this specific point. First the stress decreases, but then the effect of the inertial loads becomes dominating and the stress begins to increase.



Figure 13-12: von Mises stress as function of rotational speed at point (0.0179, 0.0535).

Modeling in COMSOL Multiphysics

To build the geometry, and especially the shape of the holes, use the tangent drawing tool.

When solving, adaptive mesh refinement helps to compute accurate stresses, as the stress concentration is not known in advance.

Model Library path: COMSOL_Multiphysics/Structural_Mechanics/pulley
MODEL NAVIGATOR

I In the Model Navigator, select 2D from the Space dimension list.

2 Select COMSOL Multiphysics>Structural Mechanics>Plane Stress, then click OK.

OPTIONS AND SETTINGS

I In the **Options** menu, open the **Constants** dialog box.

2 Enter the following constants; when finished, click **OK**.

NAME	EXPRESSION	DESCRIPTION
Р	185[kW]	Power at pulley shaft
mu	0.5	Coefficient of friction between pulley and belt
beta	(170*pi/180)[rad]	Contact angle between pulley and belt
d	5[cm]	Thickness of the pulley
n	100	RPM

3 Open the **Scalar Expressions** dialog box available in the **Options>Expressions** menu and enter the following expressions:

NAME	EXPRESSION	DESCRIPTION
r	sqrt(x^2+y^2)	Radius
theta	atan2(y,x)	Angle
omega	n/1[min]*2*pi[rad]	Rotational speed of the pulley
М	P/omega	Torque at pulley shaft
Fu	M/r	Peripheral force
F2	Fu/(exp(mu*beta)-1)	Belt force in slack side
С	F2/(r*d)	Variable used in load mode
Fn	-C*exp(mu*(theta+beta))	Normal component of the belt load
Ft	<pre>mu*C*exp(mu*(theta+beta))</pre>	Tangential component of the belt load

4 Choose **Options>Axes/Grid Settings** and give axis and grid settings according to the following table. On the **Grid** page, clear the **Auto** check box to enter the grid spacing and extra grid point. When done, click **OK**.

AXIS		GRID	
x min	-0.03	x spacing	1e-2
x max	0.03	Extra x	
y min	0.03	y spacing	1e-2
y max	0.11	Extra y	4.8e-2

GEOMETRY MODELING

- Choose Draw>Specify Objects>Circle and define a circle centered at (1.2e-2, 8.5e-2) with a radius of 1.5e-2.
- 2 Draw another circle centered at (8e-3, 6e-2) with a radius of 1.2e-2.
- **3** Select both circles and click the **Mirror** button on the Draw toolbar. Use the default setting for the reflection line and click **OK**.



- 4 Click the Tangent button on the Draw toolbar.
- 5 In the Start area, select Edge 3 of COI.

6 Click the End button and select Edge 3 of CI.



- 7 Click OK.
- 8 Draw three other tangents following the previous indication but this time with the start and end edges as in the table below.

START GEOMETRY OBJECT	START EDGE	END GEOMETRY OBJECT	END EDGE
CI	2	C2	2
C2	2	CO2	2
CO2	2	COI	3

9 Press Ctrl+A to select all the geometry objects.



IO Click the **Coerce to Solid** button on the Draw toolbar.

II Click the Delete Interior Boundaries button on the Draw toolbar.

12 Press first Ctrl+C and then Ctrl+V to copy and paste the geometry.

I3 Click **OK** in the **Paste** dialog box.

14 Click Rotate and enter -45 in the α edit field for the rotation angle.

I5 Click OK.



I6 Select both geometries, copy and paste them at the same location, and rotate the new geometries with an angle of -90° .



17 Copy and paste the new geometries and rotate them with an angle of -90° .

18 Repeat the last step once.

I9 Click **Zoom Extents**.

20 Draw two circles centered at the origin with the radii 0.025 and 0.125, respectively.

2I Click **Zoom Extents**.

Next, split the right side of the exterior boundary to model the belt contact boundary (the belt and the pulley are in contact over an angle of 170°).

- I Draw a line from (0, 0.11) to (0, 0.13).
- **2** Click the **Rotate** button and type -5 in the α edit field for the rotation angle.
- **3** Draw a line from (0, -0.11) to (0, -0.13).
- **4** Click the **Rotate** button and type **5** in the α edit field for the rotation angle.
- **5** Select the outer circle C2, and the two lines B1 and B2.
- 6 Click the Coerce to Solid button.
- **7** Select all objects (press Ctrl+A).

8 Click the Difference button.



PHYSICS SETTINGS

Subdomain Settings

- I In the Physics menu, open the Subdomain Settings dialog box.
- 2 On the Material page, click Load.
- **3** Select **Aluminum** from the library material.
- 4 In the thickness edit field type d.
- 5 Click the Load tab.
- 6 Type r*omega^2*rho_ps*cos(theta) in the Body load x dir. edit field and r*omega^2*rho_ps*sin(theta) in the Body load y dir. edit field.

This step is a transformation from cylindrical coordinates to the Cartesian coordinate system by entering the components of the radial load in the x and y directions.

- 7 Click the Body load is defined as force/volume using the thickness button.
- 8 Click OK.

Boundary Settings

I Open the Boundary Settings dialog box from the Physics menu.

2 Apply the following boundary conditions:

BOUNDARY	57, 58, 62, 6	3	
Page	Constrai	nt	
	R _×	0	
	R _y	0	
BOUNDARY	67, 68	67, 68	
Page	Load	Load	
	F _x	F _x Fn*cos(theta)-Ft*sin(theta)	
	Fy	F _y Fn*sin(theta)+Ft*cos(theta)	

3 Click the Edge load is defined as force/area using the thickness button.

4 Click OK.

MESH GENERATION

You can expect the rate of change of stresses in the corner region to be high. Use an adaptive mesh algorithm to get a more suitable mesh size according to the stress concentration. It is therefore not necessary to start with a fine mesh. COMSOL Multiphysics automatically detects areas with a high rate of change of stresses and refines them.



Click the Initialize Mesh button on the Main toolbar.

The total number of degrees of freedom should be around 11,460 (2656 elements).

COMPUTING THE SOLUTION

The rotational speed is the parameter of the parametric solver and varies from 1000 to 9000 rpm in steps of 500 rpm.

- I Open the Solver Parameters dialog box in the Solve menu.
- 2 Select Parametric from the Solver list.
- **3** Type n in the **Parameter name** edit field.
- 4 Type 1e3:5e2:9e3 in the Parameter values edit field.
- **5** Select **Direct (SPOOLES)** in the **Linear system solver** list. The system matrix becomes symmetric, and this solver can make use of that symmetry to save memory.

Use the adaptive mesh algorithm to refine the mesh during the calculation:

- 6 Select the Adaptive mesh refinement check box.
- 7 Click the Adaptive tab and type 1.3 in the Increase number of elements by edit field.
- 8 Click OK.
- 9 Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

The default plot gives the von Mises stress distribution for the highest rotational speed (9000 rpm).



- To visualize the results for different values of the rotational speed, open the Plot
 Parameters dialog box from the Postprocessing menu and use the solution for the desired rpm value by selecting from the Parameter value list.
- **2** To get a line plot of the von Mises stress at a specific point as a function of the rotational speed, open the **Cross-Section Plot Parameters** dialog box.
- 3 On the General page, select all the solutions to plot from the Solutions to use list.
- **4** In the **Point** page, enter the coordinate of the point (0.019, 0.054).
- 5 Click OK.
- **6** Figure 13-12 shows the resulting plot.

A Note About Adaptive Mesh Refinement

During the calculation the solver refines the mesh several time. You can verify this by looking at the message log.



There are two mesh refinements: The first refinement gives about 19,500 DOFs, and the second refinement gives the final solution with about 31,200 DOFs.

COMICE MUR 20 0 0 40 5 1 to to to ne Street (ps) 0.12 0.11 0.1 0.09 0.05 0.07 0.06 0.048 0.03 0.02 0.01 .0.01 -0.02 -0.03 -0.04 -0.05 -0.06 -0.0 -0.1 -0.11 -0.12 0.04 0.06 0.08 of degrees of freedom solved for: 20738 of degrees of freedom solved for: 33420 on time: 9,796 s Memory: (633 / 891) 0.1.-0.0927 EQUAL

Go to Mesh mode to see the refined mesh refined after the analysis.

The adaptive mesh algorithm, when coupled with a parametric analysis, only refines the mesh on the results based on the last parameter. In this specific case the variation of the gradient of the variable is not important enough to consider different meshes for the different parameters. In case of large variations of the gradients, we recommend to proceed using a manual mesh refinement in order to have a mesh that is suitable for all parameters.

Stiffness Analysis of a Communication Mast's Diagonal Mounting

Introduction

Communication masts usually have a framework with a bolted triangular lattice design as illustrated in Figure 13-13. The diagonals of the framework are assembled from several parts and welded together.

When operating under a given wind load at a specific location, the antenna's total rotation angle should stay below a certain limit to ensure uninterrupted communications. For the type of mast in this model, the engineers have determined that its torsional stiffness is too low, and this effect is due to the geometry of the diagonal mountings. The goal is to increase the stiffness of such a diagonal mounting by first analyzing an existing 3D CAD geometry followed by an update of the geometry and a new analysis.

The model demonstrates the import of 3D CAD assemblies into COMSOL Multiphysics from a file and through a live connection to the SolidWorks CAD



software. This case also demonstrates the individual meshing of parts with the interactive meshing tool.

Figure 13-13: Mounting detail of a mast diagonal.

Model Definition

Start by importing and analyzing the diagonal mounting CAD assembly. The assembly includes only a short section of the diagonal tubing together with the other parts of the mounting as illustrated in Figure 13-13. Although a symmetry exists in both the geometry and load for this problem, this example models the entire assembly for illustrative purposes.

The geometry file is included in the model folder (models/COMSOL_Multiphysics/ Structural_Mechanics in the COMSOL installation directory) both as a COMSOL Multiphysics binary file (mast_diagonal_mounting_pt10_mt10.mphbin) and as a Parasolid binary file (mast_diagonal_mounting_pt10_mt10.x_b). Importing the latter file requires the CAD Import Module.

After obtaining the stiffness of the assembly, assume that the geometry has been updated to improve the stiffness. Originally 10 mm, the plate thickness and mount thickness (see Figure 13-13) have been changed to 12 mm and 15 mm, respectively. You import the updated geometry file for the new analysis. These files are also included in the model folder with the name mast_diagonal_mounting_pt12_mt15 with the

extensions .mphbin and .x_b for the COMSOL Multiphysics and the Parasolid binary format, respectively.

A third approach requires that both the CAD Import Module and SolidWorks software be installed on your computer. Here you use the SolidWorks live connection in COMSOL Multiphysics to import and update the geometry. This method has the advantage that the boundary settings are saved between geometry updates. For complex problems this can result in considerable time savings. The necessary SolidWorks assembly and part files are included in the model folder; these are mast_diagonal_mounting.sldasm, mount.sldprt, plate.sldprt, and tube.sldprt.

MATERIAL PROPERTIES

Assume that the material is a structural steel. This is the default material with a Young's modulus of $2.0 \cdot 10^{11}$ N/m² and a Poisson's ratio of 0.33.

BOUNDARY CONDITIONS

Figure 13-14 shows the boundaries with an applied load and constrained displacements. Assume that the diagonal is loaded in tension by a force, F = 30 kN, which is transferred through the bolt to the mounting.



Figure 13-14: Boundaries with constrained displacements and applied loads.

Neglect contact conditions between the bolt and the mounting hole, and also neglect the constraint imposed on the mount by the bolt. Assume that the bolt fills out the entire hole volume. The load is distributed on the appropriate halves of the hole surfaces according to

$$p = \frac{F}{2(2r_{mh}b_{mh})} \cdot \frac{3}{2} \left(1 - \left(\frac{y}{r_{mh}}\right)^2\right)$$

where r_{mh} and b_{mh} are the hole's radius and the thickness, respectively.

THE FINITE ELEMENT MESH

Next use the interactive meshing tool to individually mesh different parts of the diagonal mounting assembly. This way you can use both structured and unstructured meshes with various sizes on the different parts.

During the meshing procedure keep this in mind: for the identity boundary pairs, the destination boundary should have a finer mesh than the source boundary. The finite-element mesh consists of roughly 9000 tetrahedral elements and about 250 prism elements, giving a total number of approximately 50,000 degrees of freedom. Figure 13-15 shows a 3D view of the mesh.



Figure 13-15: Finite-element mesh of the diagonal mounting assembly.

Results

Figure 13-16 shows the deformed shape of the diagonal mounting assembly with a boundary plot of the displacement in the axial direction of the diagonal. Calculate the stiffness of the mounting with the equation

$$S = \frac{F}{d_z}$$

where d_z is the axial displacement at the midplane of the holes, and F is the applied load.

Compare this formula for the stiffness to an ideal stiffness, which would result if the diagonal tube were welded to the framework and placed under the same load. Calculate the ideal stiffness with the equation

$$S_{id} = \frac{E\pi(R_{tb}^2 - r_{tb}^2)}{L}$$

where r_{tb} and R_{tb} are the tube's inner and outer radii, and L is the total length measured in the axial direction up to the mid-plane of the mount holes. For the material under study, E = 200 GPa is the Young's modulus.

The stiffness ratio, $S_R = S/S_{id}$ is 0.42 for the case when the plate thickness and the mount thickness are 10 mm. With the updated geometry (plate thickness set to 12 mm and the mount thickness set to 15 mm), the stiffness ratio increases to 0.59.



Figure 13-16: Deformed shape and boundary plot of the axial displacement for the mounting assembly with an end-plate thickness of 10 mm and mount thickness of 10 mm.

Model Library path: COMSOL_Multiphysics/Structural_Mechanics/ mast_diagonal_mounting_mphbin, mast_diagonal_mounting_parasolid, mast_diagonal_mounting_swlive

Modeling using the Graphical User Interface

MODEL NAVIGATOR

- I In the Model Navigator go to the New page and select 3D from the Space dimension list.
- 2 Select COMSOL Multiphysics>Structural Mechanics>Solid, Stress-Strain>Static analysis.
- 3 Click OK to close the Model Navigator.

IMPORTING THE CAD GEOMETRY

Multiphysics Geometry File

- I On the File menu select Import>CAD Data From File.
- 2 In the Files of type list select COMSOL Multiphysics file.
- **3** Select the file mast_diagonal_mounting_pt10_mt10.mphbin.
- 4 Click Import.

Parasolid File

With the CAD Import Module you might also select the Parasolid file for import.

- I On the File menu select Import>CAD Data From File.
- 2 On the Files of type list select Parasolid file.
- **3** Select the file mast_diagonal_mounting_pt10_mt10.x_b.
- 4 Click Import.

SolidWorks Live Connection

With the CAD Import Module and SolidWorks software you might want to base the analysis on the SolidWorks assembly.

Have the file mast_diagonal_mounting.sldasm open in SolidWorks and carry out the following step in COMSOL Multiphysics:

On the **File** menu select **SolidWorks Connection>Initialize** to import the assembly into COMSOL Multiphysics.

Creating Boundary Pairs

I Make sure all assembly parts are selected; if not, press Ctrl+A to select all parts.

The next step identifies the identity boundary pairs and creates the necessary identity constraints for the quantities that are equal on these boundaries.

2 Click the Create Pairs button on the Draw toolbar.

It is important that the source and destination domains be correctly defined for the identity pairs in the assembly. The destination should have a finer mesh than the source. For some of the identified identity boundaries you must interchange the source and destination so that they agree with the mesh you generate later on.

- 3 From the Physics menu select Identity Pairs>Identity Boundary Pairs.
- 4 In the Identity pairs list select Pair I.
- **5** Click the **Interchange Source and Destination** button.

- 6 Repeat the previous two steps for Pair 2 and Pair 3.
- 7 Click OK to close the Identity Boundary Pairs dialog box.

OPTIONS AND SETTINGS

Constants

- I On the **Options** menu select **Constants**.
- 2 Enter a variable with the Name Force and an Expression 30[kN].
- 3 Click OK.

Expressions

- I On the **Options** menu select **Expressions>Global Expressions**.
- 2 Enter expressions as given in the following table. Note that the last two expressions are for the load components; they specify both the magnitude and direction of the load that is applied in the opposite direction to the surface normal.

NAME	EXPRESSION
mount_th_ac	mount_pos1-mount_pos2
mounthole_xarea	mount_th_ac*2*mounthole_r
tube_xarea	pi*(tube_R^2-tube_r^2)
Stiffness_ideal	2e11*tube_xarea/L_tot
Stiffness_current	Force/z_displ
Stiffness_ratio	Stiffness_current/Stiffness_ideal
mounthole_Fy	-Force/2/mounthole_xarea*3/2*(1-(y/ mounthole_r)^2)*ny
mounthole_Fz	-Force/2/mounthole_xarea*3/2*(1-(y/ mounthole_r)^2)*nz

3 Click OK.

Integration Variables

Use integration coupling variables to make globally available the geometry dimensions as well as the displacement of a point.

- I On the Options menu select Integration Coupling Variables>Point Variables.
- **2** From the **Point selection** list choose 25.

3 Enter an expression with the **Name** tube_r and an **Expression** x.



4 Repeat the previous two steps for the following points and the corresponding expressions.

POINT SELECTION	NAME	EXPRESSION
27	tube_R	x
44	mount_pos2	x
50	L_tot	z
50	mounthole_r	У
50	z_displ	w
50	mount_pos1	x

5 Click OK.

PHYSICS SETTINGS

Boundary Conditions

- I On the Physics menu select Boundary Settings.
- 2 On the Boundaries page locate the Boundary selection list and choose 22 and 30.

- 3 On the Load page enter mounthole_Fy in the Fy edit field and enter mounthole_Fz in the Fy edit field.
- 4 From the **Boundary selection** list choose 13.
- 5 On the **Constraint** page select the **R**_x, **R**_y, and **R**_z check boxes. Make sure their values are set to zero.
- 6 Click OK.

MESH GENERATION

Generate the mesh individually for the parts using the interactive meshing tool. Start by meshing the parts that should have an unstructured mesh by using the free mesher.

- I On the Mesh menu select Free Mesh Parameters.
- 2 On the Global page locate the Predefined mesh sizes list and select Fine.
- **3** On the **Subdomain** page select Subdomains 3 and 4.
- 4 Click the Mesh Selected button.
- 5 On the Global page find the Predefined mesh sizes list and select Finer.
- 6 On the Subdomain page select Subdomain 1.
- 7 Click the Mesh Selected button. Click OK.

Now use the swept mesher on the tube.

- I On the Mesh menu select Swept Mesh Parameters.
- 2 From the Predefined mesh sizes list select Fine.
- **3** From the **Subdomain selection** list select **2**.
- 4 Click the Mesh Selected button.
- 5 Click OK.

Figure 13-15 shows the meshed assembly.

COMPUTING THE SOLUTION

- I From the Solve menu select Solver Parameters.
- 2 On the Stationary page locate the Linearity list and select Linear.
- 3 Click OK.
- 4 Click the **Solve** button on the Main toolbar to start the analysis.

POSTPROCESSING AND VISUALIZATION

I On the Postprocessing menu select Plot Parameters.

- 2 On the General page go to the Plot type area. Clear the Slice check box and select the Boundary and Deformed shape check boxes.
- 3 On the Boundary page go to the Predefined quantities list and select z-displacement.
- 4 Click **OK** to close the dialog box and to display the plot in Figure 13-16 on page 542.
- 5 On the Postprocessing menu select Data Display>Global.
- 6 From the Predefined quantities list select Global Expressions>Stiffness_ratio.
- **7** Click **OK**. The value of the stiffness ratio, 0.42, appears in the message log at the bottom of the user interface.

UPDATE OF CAD GEOMETRY

Click the **Draw Mode** button on the Main toolbar.

The following steps describe how to import the CAD geometry from a COMSOL Multiphysics geometry file and from a Parasolid file.

Multiphysics Geometry File

- I Press Delete to delete the current geometry (first make sure that the geometry is selected).
- 2 On the File menu select Import>CAD Data From File.
- 3 In the Files of type list select COMSOL Multiphysics file.
- **4** Select the file mast_diagonal_mounting_pt12_mt15.mphbin.
- 5 Click Import.

Parasolid File

- I Press Delete to delete the current geometry.
- 2 On the File menu select Import>CAD Data From File.
- 3 On the Files of type list select Parasolid file.
- **4** Select the file mast_diagonal_mounting_pt12_mt15.x_b.
- 5 Click Import.

SolidWorks Live Connection

Now you can control the plate and mount thicknesses of the SolidWorks assembly from COMSOL Multiphysics. This is possible if you define constants that have the same name as the variables defining these dimensions in the SolidWorks file.

- I On the **Options** menu select **Constants**.
- 2 Enter a variable with the Name plate_thickness and an Expression 12e-3.

- 3 Enter a variable with the Name mount_thickness and an Expression 15e-3.
- 4 Click OK.
- **5** On the **File** menu select **SolidWorks Connection>Update** to update and import the assembly into COMSOL Multiphysics.

Creating Boundary Pairs

Repeat the steps under "Creating Boundary Pairs" on page 543 only if you imported the COMSOL Multiphysics geometry or Parasolid file. Boundary pairs are retained with the SolidWorks live connection and do not need to be specified again.

OPTIONS SETTINGS

Integration Variables

The names of the variables are retained. Select the points and enter the corresponding expressions as described in "Integration Variables" on page 544.

PHYSICS SETTINGS

Boundary Conditions

Repeat the steps under "Boundary Conditions" on page 545 only if you imported the COMSOL Multiphysics geometry or Parasolid file. Boundary conditions are retained with the SolidWorks live connection and need not be specified again.

MESH GENERATION

Repeat the steps under "Mesh Generation" on page 546.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar to start the analysis.

POSTPROCESSING AND VISUALIZATION

- I On the Postprocessing menu select Data Display>Global.
- 2 From the Predefined quantities list select Global Expressions>Stiffness_ratio.
- **3** Click **OK**. The value of the stiffness ratio, 0.59, appears in the message log at the bottom of the user interface.

3D Analysis of a Formula One Car Front Wing

Introduction

The bodywork of Formula One (F1) cars is designed based on two important criteria: a car's aerodynamics, and technical regulations (Ref. 1) set forth by the Federation Internationale de l'Automobile (FIA). These regulations specify a maximum allowable downforce, limits on body dimensions, as well as specific tests for body flexibility.

The flexibility test from Section 3.17.2 of the regulations dictates that a load be applied to an F1 car's front wing with a load adaptor that has a specified geometry. The maximum deflection should be less than 10 mm.

This example models this test on a geometry that is based on the front wing of a Ferrari F1 2004 (F104001) as illustrated in Figure 13-17. The goal of this model is to demonstrate the use of assemblies in combination with interactive meshing while analyzing a body-flexibility test performed on F1 cars.



Figure 13-17: Formula One car and model of the front-wing assembly.

Model Definition

Assume that the problem is symmetric so you need to model only half of the front wing. Neglect contact phenomena between the load adaptor and the wing surface. A geometry file, f1_front_wing.mphbin, is included in the model folder

(models/COMSOL_Multiphysics/Structural_Mechanics) and consists of an assembly of several parts including the load adaptor (see Figure 13-18).



Figure 13-18: 3D CAD geometry of the front-wing assembly with the load adaptor for the flexibility test.

MATERIAL PROPERTIES

For the material in the front wing, assume isotropic elasticity with constitutive parameters and a density as in this table:

PROPERTY	VALUE
Young's modulus	0.8·10 ¹¹ N/m ²
Poisson's ration	0.25
Density	1700 kg/m ³

For the material in the load adaptor assume a Young's modulus of $2.0 \cdot 10^{11}$ N/m² and a Poisson's ratio of 0.33.

BOUNDARY CONDITIONS

Figure 13-14 shows the boundaries with an applied load or constrained displacements. Apply a load of -500 N on the surface of the load adaptor. You also constrain displacements in all directions at the upper surface of the support that connects the

wing to the rest of the body. The mid plane along the F1 car is a symmetry plane. Constrain displacement in the x direction for all surfaces on this plane.



Figure 13-19: Boundaries with constrained displacements or applied load in the model.

THE FINITE ELEMENT MESH

Use the interactive meshing tool to individually mesh different parts of the wing assembly. This way you can use both a structured and an unstructured mesh with various sizes on the different parts.

For some of the parts—including the wings, parts of the air guide underneath the wings, and the support that fixes the assembly to the rest of the body—generate a structured mesh with prism elements. Achieve this with the swept mesher. Generate an unstructured mesh on the remaining parts by using the free mesher.

During the meshing procedure keep in mind that for the identity boundary pairs the destination boundary should have a finer mesh than the source boundary.

The finite-element mesh consists of approximately 49,000 tetrahedral elements and roughly 3,000 prism elements, giving a total number of approximately 300,000 degrees of freedom. Figure 13-20 shows a 3D view of the mesh.



Figure 13-20: Finite-element mesh of the front-wing assembly with the load adaptor.

Results

The deformed shape of the wing assembly with a boundary plot of the vertical (y-) displacement appears in Figure 13-21.



Figure 13-21: Deformed shape and boundary plot of the y-displacement of the wing assembly.

It is evident from the figure that the maximum vertical displacement occurs at the back outer corner of the wing's end plate. To find its value, evaluate the *y*-displacement at the point marked in Figure 13-22. The maximum value of -2.2 mm is well below the allowed deflection of -10 mm. It is important to realize, however, that this analysis shows a lower value for the *y*-displacement when compared to a flexibility test on an actual F1 car because in that case the rest of the body would also contribute to the

deflection. Another factor that influences the results is the choice of constitutive parameters.



Figure 13-22: Location of the maximum vertical deflection of the wing assembly.

Reference

1. http://www.fia.com/resources/documents/ 1603301296_2006_F1_TECHNICAL_REGULATIONS.pdf

Model Library path:

COMSOL_Multiphysics/Structural_Mechanics/f1_front_wing

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

I In the Model Navigator go to the New page, then select 3D from the Space dimension list.

2 In the list of application modes select

COMSOL Multiphysics>Structural Mechanics>Solid, Stress-Strain>Static analysis.

3 Click OK.

IMPORT OF CAD GEOMETRY

- I On the File menu select Import>CAD Data From File.
- 2 In the Files of type list select COMSOL Multiphysics file.
- 3 Browse to the COMSOL34/models/COMSOL_Multiphysics/ Structural_Mechanics directory and select the file f1_front_wing.mphbin.
- 4 Click Import.
- 5 Make sure all the assembly parts are selected; if not press Ctrl+A to select all parts.
- 6 On the Draw menu select Use Assembly.

The next step identifies the identity boundary pairs and creates the necessary identity constraints for the quantities that are equal on these boundaries.

7 Click the **Create Pairs** button on the Draw toolbar.

It is important that the source and destination domains are correctly defined for the identity pairs in the assembly. The destination should have a finer mesh than the source. For some of the identified identity boundaries you must interchange the source and destination so they agree with the mesh you generate later on.

- 8 From the Physics menu select Identity Pairs>Identity Boundary Pairs.
- 9 In the Identity pairs list select Pair 5.
- **IO** Click the **Interchange Source and Destination** button.
- II Repeat the previous two steps for each of the following pairs: **Pair 16**, **Pair 17**, and **Pair 18**.
- I2 Click OK.

PHYSICS SETTINGS

Subdomain Settings

- I On the Physics menu select Subdomain Settings.
- 2 In the Subdomain selection list select Subdomains 1 through 14.
- 3 Enter 0.8e11 in the E (Young's modulus) edit field, 0.25 in the ν (Poisson's ratio) edit field, and 1700 in the ρ (density) edit field.
- 4 Click OK.

Boundary Conditions

- I On the Physics menu select Boundary Settings.
- 2 On the Boundaries page locate the Boundary selection list and select Boundary 182.
- 3 On the Load page type -2.55e5 in the F_y edit field.
- 4 In the Boundary selection list choose Boundary 27.
- 5 On the **Constraint** page select the **R**_x, **R**_y and **R**_z check boxes. Make sure their values are set to zero.
- 6 In the Boundary selection list choose Boundaries 1, 5, 10, 14, and 19.
- 7 On the **Constraint** page select the $\mathbf{R}_{\mathbf{x}}$ check box. Make sure its value is set to zero.
- 8 Click OK.

MESH GENERATION

Generate the mesh individually for the parts using the interactive meshing tool. Start by meshing the parts that should have an unstructured mesh by using the free mesher.

- I On the Mesh menu select Free Mesh Parameters.
- 2 On the Global page find the Predefined mesh sizes list and select Coarse.
- 3 On the Subdomain page choose Subdomains 11 and 15.
- 4 Click the Mesh Selected button.
- **5** Repeat the previous three steps according to the following table.

SUBDOMAIN	PREDEFINED MESH SIZE
2, 4, 7, 8	Normal
12	Fine

At this point the mesh should look like the one in the following figure:



In the next steps you mesh one edge of each of the three main wings. By doing so before meshing the corresponding subdomains using the swept mesher you can have a structured mesh that is coarse in the cross section but still resolves the curvature of the edges along the wing.

- I On the Global page locate the Predefined mesh sizes list and select Extra fine.
- 2 On the Edge page select Edges 3, 20 and 42.
- **3** Click the **Mesh Selected** button.
- 4 Click OK to close the Free Mesh Parameters dialog box.

Now use the swept mesher on the wings and the other remaining parts.

- I On the Mesh menu select Swept Mesh Parameters.
- 2 In the Predefined mesh sizes list select Normal.
- **3** In the **Subdomain selection** list select Subdomain 1.
- **4** Click the **Sweep Direction** tab.
- **5** Select the Manual specification of sweep direction check box.
- 6 Enter 4 in the Source face edit field.
- 7 Enter 1 in the Target face edit field.
- 8 Click the Mesh Selected button.
- 9 Click on the Element Layers tab.

10 Repeat Steps 2 to 9 above for the subdomains listed in the following table

SUBDOMAIN	PREDEFINED MESH SIZES	SOURCE FACE	TARGET FACE
3	Normal	13	10
5	Finer	23	19

II In the Predefined mesh sizes list select Normal.

12 In the Subdomain selection list select Subdomains 6, 9, 10, 13, and 14.

I3 Click the **Mesh Selected** button.

I4 Click OK.

Figure 13-20 shows the finished mesh of the assembly.

COMPUTING THE SOLUTION

- I On the Solve menu select Solver Parameters.
- 2 On the Stationary page go to the Linearity list and select Linear.
- 3 Click OK to close the Solver Parameters dialog box.
- 4 Click the **Solve** button on the Main toolbar to start the analysis.

POSTPROCESSING AND VISUALIZATION

- I On the Postprocessing menu select Plot Parameters.
- 2 On the General page go to the Plot type area. Clear the Slice check box, then select the Boundary and Deformed shape check boxes.
- 3 On the Boundary page find the Predefined quantities list and select y-displacement.
- 4 Click **OK** to close the dialog box and to display the plot in Figure 13-21 on page 553.
- 5 On the Postprocessing menu select Point Evaluation.
- 6 In the Point selection list select Point 246.
- 7 In the Predefined quantities list select y-displacement.
- 8 Click **OK** to close the dialog box. The displacement value (-2.2 mm) appears in the message log at the bottom of the user interface.

14

Wave Propagation Models

Wave propagation models describe the propagation of electromagnetic waves, light, fluids, and more. For additional wave propagation models, see the chapters containing acoustics models, electromagnetics models, and fluid dynamics models.

Diffraction Patterns

Introduction

This example resembles a two-slit interference experiment with water waves or sound. The model mimics the plane-wave excitation with two thin waveguides leading to slits in a screen and computes the diffraction pattern on the other side of the screen.

Model Definition

Theory predicts amplitude minima along rays where the difference in travel distance is an odd multiple of half the wavelength, and maxima at even multiples. For $n = 0, \pm 1, \pm 2, ...$

$$\begin{cases} \min, \sin \theta = \left(n + \frac{1}{2}\right) \frac{\lambda}{D} \\ \max, \sin \theta = n \frac{\lambda}{D} \end{cases}$$

In this model, the distance *D* between the slits is 2λ . Maxima should then be present at $\theta = 0^{\circ}$ and 30° , while minima should appear at $\theta = 14.48^{\circ}$ and 48.59° .

Equation

For time-harmonic propagation, the wave equation turns into the Helmholtz equation:

$$-\nabla \cdot (\nabla u) - k^2 u = 0, \qquad k = \frac{2\pi}{\lambda}$$

Boundary Conditions

The absorbing boundary conditions have the form

$$\mathbf{n} \cdot (\nabla u) + iku = 2ik, \text{ inflow} \\ \mathbf{n} \cdot (\nabla u) + iku = 0, \text{ outflow}.$$

Results



The plot above shows the diffraction pattern clearly. The effect of quantization is that the numerical wavelength differs from λ , which results in a shift of the angles. You can correct for this effect by adjusting the value of k in the Helmholtz equation to the element size. These practices are important for modeling the interference effects of monochromatic waves.

Model Library path:

COMSOL_Multiphysics/Wave_Propagation/diffraction_patterns

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select 2D in the Space dimension list.
- 2 In the list of application modes, open COMSOL Multiphysics>PDE Modes and then PDE, Coefficient Form.

- **3** Select **Stationary analysis**. Make sure that **Lagrange Quadratic** elements are selected in the **Element** list.
- 4 Click OK.

OPTIONS AND SETTINGS

- I Open the Axes/Grid Settings dialog box.
- 2 Set the axes and grid settings according to the following table:

AXIS		GRID	
x min	-0.75	x spacing	0.5
x max	2.25	Extra x	
y min	- 1	y spacing	0.5
y max	1	Extra y	-0.015 0.015

3 Open the **Constants** dialog box.

4 Enter the following constants (the descriptions are optional):

NAME	EXPRESSION	DESCRIPTION
1	0.1	Wavelength (m)
k	2*pi/l	Wave number (rad/m)

GEOMETRY MODELING

- I Draw a circle with its center at (0.5, 0) and a radius of 1.
- **2** Draw a rectangle centered at (0, 0) with sides of 1 and 2 units for the x and y direction, respectively.
- 3 Select both objects and click the Difference button to create a half circle.
- **4** Draw a rectangle with its lower left corner at (0, -0.015) and upper right corner at (0.5, 0.015).
- 5 Click the **Move** button and set the *y*-displacement to 0.1. Leave the *x*-displacement at 0.
- 6 Click OK.
- 7 Duplicate and place the rectangle by pressing Ctrl+C and then Ctrl+V.
- 8 Click the **Move** button again and set the *x* and *y*-axis displacements to 0 and -0.2, respectively.
- 9 Select all objects.

10 From the Draw menu, choose Create Composite Object.
II In the Set formula edit field, type C01+R1+R2.

12 Clear the **Keep interior boundaries** check box.

I3 Click OK.

14 Click the Zoom Extents button on the Main toolbar.



PHYSICS SETTINGS

Boundary Conditions

- I From the Physics menu, choose Boundary Settings.
- 2 In the **Boundary Settings** dialog box, enter these boundary conditions (i denotes the imaginary unit):

SETTINGS	BOUNDARIES I, 4 (INFLOW)	BOUNDARIES 2, 3, 5–9	BOUNDARIES 10, 11 (OUTFLOW)
Туре	Neumann	Neumann	Neumann
q	i*k	0	i*k
g	2*i*k	0	0

Subdomain Settings

I From the Physics menu, choose Subdomain Settings.

2 In the Subdomain Settings dialog box, enter the following PDE coefficients:

SETTINGS	SUBDOMAIN I		
c	1		
a	- k^2		
f	0		

MESH GENERATION

- I In the Free Mesh Parameters dialog box, click the Global tab
- 2 Click the **Custom mesh size** button, then enter 0.03 in the **Maximum element size** edit field (approximately 3 second-order elements per wavelength).
- 3 Click the **Remesh** button, then click **OK**.

COMPUTING THE SOLUTION

Click the Solve button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

By default, COMSOL Multiphysics plots the real part of the solution using the jet colormap. To make the diffraction pattern appear even more clearly, change to the wave colormap as follows:

- I Click the Plot Parameters button on the Main toolbar.
- 2 On the Surface page, select wave from the Colormap list in the Surface color area.
- 3 Click OK.

Modeling Using the Programming Language

I Initialize the FEM structure:

```
clear fem
fem.shape = 2;
fem.sshape = 2;
fem.const = {'k',2*pi/0.1};
```

2 Create the problem geometry:

c1 = circ2(0.5,0,1); r1 = rect2(-0.5,0.5,-1,1); r2 = rect2(0,0.5,0.085,0.115); r3 = rect2(0,0.5,-0.115,-0.085); fem.geom = (c1-r1)+r2+r3;

3 Define the boundary conditions:

fem.bnd.q = {'i*k',0,'i*k'}; fem.bnd.g = {'2*i*k',0,0}; fem.bnd.ind = [1 2 2 1 2 2 2 2 2 2 3 3];

4 Specify the PDE coefficients:

fem.equ.c = 1; fem.equ.a = $'-k^2'$;

5 Initialize the mesh using a maximum element size of 0.03:

fem.mesh = meshinit(fem, 'hmax', 0.03);

6 Solve the problem using the stationary solver:

```
fem.xmesh = meshextend(fem);
fem.sol = femstatic(fem);
```

7 Plot the real part of the complex solution:

```
postplot(fem,'tridata','u','axisequal','on','trimap','wavemap')
```

8 Generate an animation:

```
postmovie(fem,'tridata','u','axisequal','on')
```

15

Benchmark Models

Benchmark models are those with known results from experiments and measurements, from analytical solutions and formulas, or from established benchmark problems published by organizations such as NAFEMS. Many of the other examples in the Model Library and the COMSOL Multiphysics modules are benchmark models as well. Most introductory models in the *COMSOL Multiphysics Modeling Guide* compare results to theoretical values or established benchmark data.

Isospectral Drums

Introduction

This example examines an interesting question posed by Mark Kac in 1966 (Ref. 1): "Can one hear the shape of a drum?"

Striking a drum excites a spectrum of vibration modes that together make up the instrument's characteristic sound or acoustic signal. These vibration modes correspond to the eigenmodes, or eigenfunctions, of the drum's membrane. Thus you can study this problem by solving eigenvalue problems for stretched membranes.

If you can find two differently shaped membranes that have identical eigenvalues—in other words, they are *isospectral*—then it is not possible to hear the shape of a specific drum.

In 1992, Gordon, Webb, and Wolpert (Ref. 2) showed that there are indeed sets of different planar shapes (nonisometric shapes) that are isospectral.

Work by Driscoll (Ref. 3) contains the following example of two planar shapes that sound the same.

Model Definition

The model shows the eigensolutions (the eigenvalues and eigenmodes) in two isospectral domains. For both cases, use the solution to the same eigenvalue PDE:

 $-\Delta u = \lambda u$

The membranes are fixed at the boundaries, that is, a homogeneous Dirichlet boundary condition for all boundaries.

Results

The eigenvalues show that the domains are isospectral. This is one of the eigenmodes for the first domain:



The eigenvalues are the same for the other domain. The sign of the eigenfunction is arbitrary and can vary from case to case:



Using Richardson extrapolation, the results for the first six eigenvalues of the two domains agree to at least five decimal places with the established values in Ref. 3.

Another postprocessing step shows that the eigensolutions are orthogonal by integrating the product of two different eigensolutions.

Modeling in COMSOL Multiphysics

Build two COMSOL Multiphysics models that solve the eigenvalue PDE on two different 2D domains and compare the sets of eigenvalues. To achieve higher accuracy in the results, use *Richardson extrapolation*, which increases the accuracy if you know the behavior of the numerical error.

The model also shows hot to use the with operator to access different eigenmodes during postprocessing.

References

1. M. Kac, "Can one hear the shape of a drum?," *American Math. Mon.*,73 Part II (1966), pp. 1–23.

2. C. Gordon, D. Webb, and S. Wolpert, "Isospectral plane domains and surfaces via Riemannian orbifolds," *Invent. Math.*, 110 (1992), pp. 1–22.

3. T. Driscoll, "Eigenmodes of isospectral drums," *Technical Report-Center for Theory and Simulation in Science and Engineering*, Cornell University, Ithaca, N.Y., CTC95TR209, May 1995.

Model Library path: COMSOL_Multiphysics/Benchmarks/isospectral_drum1

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

- I Select 2D in the Space dimension list.
- In the list of application modes, open COMSOL Multiphysics>PDE Modes and then PDE, Coefficient Form. Select Eigenvalue analysis. Make sure Lagrange - Quadratic elements are selected.
- 3 Click OK.

OPTIONS AND SETTINGS

- I From the Options menu, select Axes/Grid Settings.
- 2 Click the Axis tab.
- 3 Clear the Axis equal check box and enter axis limits from the table below.
- **4** Click the **Grid** tab.
- 5 Clear the Auto check box and enter grid line from the table below
- 6 Click OK.

AXIS		GRID	
x min	- 4	x spacing	1
x max	4	Extra x	
y min	- 3	y spacing	1
y max	3	Extra y	

GEOMETRY MODELING-THE FIRST ISOSPECTRAL DRUM

- Draw a series of lines between the following points: (-3, -3), (-3, -1), (1, 3), (1, 1), (3, 1), (1, -1), (-1, -1), and (-1, -3).
- **2** Close the polygon by clicking the right mouse button.

PHYSICS SETTINGS

Boundary Conditions

There is no need to change the boundary coefficients. The default Dirichlet condition u = 0 on the boundary is correct. This value corresponds to clamping the drum's membrane at the edges.

```
Subdomain Settings
```

The following equation describes a general eigenvalue PDE:

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

so use the default values c = 1 and $d_a = 1$. All other coefficients are equal to 0 by default (the source term f is 1 but is not part of the eigenvalue PDE).

MESH GENERATION

Click the Initialize Mesh button on the Main toolbar to create a mesh.

COMPUTING THE SOLUTION

Solve the problem. The default settings give the six lowest eigenvalues.

POSTPROCESSING AND VISUALIZATION

Plot some of the eigenfunctions associated with the eigenvalues. Select the corresponding eigenvalue from the **Eigenvalue** list on the **General** page in the **Plot Parameters** dialog box. The plot shows the oddly shaped drum seen from above.

Model Library path: COMSOL_Multiphysics/Benchmarks/isospectral_drum2

GEOMETRY MODELING-THE SECOND ISOSPECTRAL DRUM

- Draw a sequence of lines between the following points: (-3, -1), (-3, 1), (1, 1), (1, 3), (3, 1), (1, -1), (-1, -1), (-1, -3).
- **2** Close the polygon by right clicking the mouse.

In setting boundary and PDE coefficients, follow the same procedure as for the first drum.

POSTPROCESSING

Showing that the Eigenmodes are Orthogonal

Use the with operator to access different eigenmodes in order to show that they are orthogonal:

- I From the Postprocessing menu, choose Subdomain Integration.
- 2 Select Subdomain 1.
- **3** Type with (1, u) *with (2, u) in the **Expression** edit field. This specifies an integrand that is the product of the first and the second eigenmode.
- 4 Click Apply.

The value, which should be very small (in the order of 10^{-15}), appears in the message log at the bottom of the COMSOL Multiphysics user interface. Ideally, the result should be zero when the eigenmodes are orthogonal, but using a numerical method you can expect a small nonzero number.

Note: The following section requires that you run COMSOL Multiphysics with COMSOL Script or MATLAB.

To improve the accuracy of the eigenvalues, continue with COMSOL Script or MATLAB using Richardson extrapolation. First export the model:

From the **File** menu, choose **Export FEM Structure**. Then continue with the instructions on Richardson extrapolation in the section below.

Modeling Using the Programming Language

The following sequence of commands finds the eigenvalues and the corresponding eigenfunctions for the first drum:

I Initialize the FEM structure and create the geometry and mesh:

```
clear fem
fem.shape = 2;
fem.sshape = 2;
fem.geom = poly2([-3,-3, 1, 1, 3, 1, -1, -1], ...
[-3,-1,3,1,1,-1,-1,-3]);
fem.mesh = meshinit(fem);
fem.mesh = meshrefine(fem);
```

2 The eigenvalue PDE coefficients *c* and *d* for this problem are *c* = 1 and *d* = 1. Settingh to 1 ensures zero Dirichlet conditions on the boundaries:

```
fem.equ.c = 1;
fem.equ.da = 1;
fem.bnd.h = 1;
```

3 Create the extended mesh and call the eigenvalue solver. Find the resulting eigenvalues in the field fem.sol.lambda.

```
fem.xmesh = meshextend(fem);
fem.sol = femeig(fem);
```

4 To plot the fourth eigenfunction as a surface plot, type

```
postplot(fem,'tridata','u','solnum',4,'axisequal','on');
```

5 To model the second drum, replace the line defining the geometry with the following:

```
fem.geom = poly2([-3,-3, 1, 1, 3, 1, -1, -1], ...
[-1,1,1,3,1,-1,-1,-3]);
```

RICHARDSON EXTRAPOLATION

The Richardson extrapolation part assumes that you have already created an FEM structure, either by exporting from COMSOL Multiphysics or by following the command-line instructions.

- I To begin, export or create the FEM structure for the first drum.
- **2** For the Richardson extrapolation, you need three meshes of different levels of refinement. To save the previous results, copy the FEM structure into three new FEM structures:

fem1 = fem; fem2 = fem; fem3 = fem;

3 First create a coarse mesh and solve the problem:

```
fem1.mesh = meshinit(fem1);
fem1.mesh = meshrefine(fem);
fem1.xmesh = meshextend(fem1);
fem1.sol = femeig(fem1);
```

The field fem1.sol.lambda now contains the first six eigenvalues.

4 Second, create an intermediate mesh by refining the coarse mesh and then solve the problem:

```
fem2.mesh = meshrefine(fem1);
fem2.xmesh = meshextend(fem2);
fem2.sol = femeig(fem2);
```

5 Now create a fine mesh by refining again and solve the problem:

```
fem3.mesh = meshrefine(fem2);
fem3.xmesh = meshextend(fem3);
fem3.sol = femeig(fem3);
```

6 Now you can extrapolate these three results. To do so you must know the error behavior. The numerical eigenvalue λ_h differs from the exact eigenvalue λ according to

$$\lambda_h \approx \lambda + c_1 h^{4/3} + c_2 h^3 + \dots$$

where h is the mesh parameter describing the length of the typical triangle edges. The exponent 4/3 appears in the error due to the occurrence of concave corners in the geometry and is not dependent on the element order. The second term corresponds to the standard error of the finite element method using second order Lagrange elements.

Because you have three meshes with the parameters h, h/2, and h/4, and because you also have the three corresponding numerical eigenvalues λ_h , $\lambda_{h/2}$, and $\lambda_{h/4}$, you can solve for the unknowns λ , c_1 , and c_2 . This procedure yields a linear system

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & (1/2)^{4/3} & (1/2)^3 \\ 1 & (1/4)^{4/3} & (1/4)^3 \end{bmatrix} \begin{bmatrix} \lambda \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \lambda_h \\ \lambda_{h/2} \\ \lambda_{h/4} \end{bmatrix}$$

that you can solve by entering the following commands:

v = [1;1/2;1/4]*ones(1,3); e = ones(3,1)*[0 4/3 4]; A = v.^e; b = [fem1.sol.lambda;fem2.sol.lambda;fem3.sol.lambda]; x = A\b lambda1 = x(1,:)

The vector lambda1 now contains the extrapolated values for the first drum. Notice that the extrapolated values are all smaller than the corresponding non-extrapolated values. This is because solving an eigenvalue problem is related to solving a certain equivalent minimization problem, and a better numerical solution typically corresponds to a lower value. You can also observe the same effect when refining the mesh (see the vector b as created above).

Now export or create the FEM structure corresponding to the second drum and repeat the steps above. For comparison, remember to rename the lambda1 vector, for example, lambda2.

Finally, compare the Richardson extrapolation results for the first six eigenvalues of the two domains and note that they agree to at least five decimal places.

2.537944	3.655510	5.175560	6.537563	7.248088	9.209311
2.537944	3.655512	5.175560	6.537560	7.248088	9.209319

Scientists have determined values for the first six eigenvalues, and they appear in the list below to approximately 12 decimal places (Ref. 3).

```
2.53794399980
3.65550971352
5.17555935622
6.53755744376
7.24807786256
9.20929499840
```

INDEX

A absorbing boundary condition 560 AC/DC Module 53 acoustic energy 11 acoustics 5 of a muffler 10 ALE 247, 414 algebraic multigrid preconditioner 63 AMG preconditioner see algebraic multigrid preconditioner anode 442 antinode 21 aperture, of fracture 270 application mode Structural mechanics 507 Weak Form, Point 313 application modes Convection and Conduction 359 Incompressible Navier-Stokes 358 arbitrary Lagrangian-Eulerian technique 247 Archimedes' force 378 Argyris element 192 integration order 195 aspect ratios avoiding large in geometries 150 axis settings 362 B backstep model using Argyris elements 192 benchmark drag coefficient 204 lift coefficient 204 benchmark models 567 bias in diode model 448 bilinear interpolation 271

bipolar junction transistor 457

Black-Scholes equation 166 Bohr radius 424 Boltzmann transport theory 442, 476 Boolean operation 59 boundary condition absorbing 560 symmetry 443 boundary integration 372 Boussinesq approximation 359, 378 bulk concentration in transport and adsorption model 31 buoyancy force 359, 378 C capacitance 93, 97 computing 94 capacitor 93 cardiac tissue signal propagation in 100 cathode 442 coefficient of friction 522 complex Landau-Ginzburg equations 101 Component Library 289 components, merging 299 compressible inviscid flow 230 COMSOL Multiphysics Subsystem 336 COMSOL Script using data from 271 using functions in 273 Constants dialog box 361 constants, defining 361 constitutive relations in electromagnetics 54 contact surface in shock-tube problems 233 Convection and Conduction application mode 359 convection-diffusion equation

in transport and adsorption model 32 coupling local 397 coupling variable 314 crankshaft 496 cross-section plot 315, 456, 492 cubic law equation 270 current density 53 cylinder flow of fluid past 198 D damping in muffler 12 de Vries 128 dest operator 120 differential algebraic system 304 differential-algebraic equation system 390 diffraction patterns 560 diffusion coefficient 270 diode 442 doping of 446 forward bias region 450 direct solver 63 discontinuous basis functions 112 discontinuous elements 111 discontinuous Galerkin method 112 dispersion tensor 261 distributed spice model of integrated bipolar transistor 457 divergence 359 divergence operator 270 divergence-free element 192 doping 442 doping function for diode 446 drag coefficient 204 dynamic viscosity 48, 324, 359 dynamical system 329

E eigenfrequencies excited by speakers 21 for empty room 20 of a room 19 eigenmodes classes of 20 of a room 19 eigenvalue analysis 496 electric charge density 53 electric displacement 53 electric energy in capacitor 93 electric field 53 electric flux density 53 electric potential 64 electromagnetics models 53 electron energy distribution function 171 element Argyris 192 error behavior for eigenvalue solution 574 Euler equations of gas dynamics 230 European stock option 166 excitable media 100 extended multiphysics 310 extrusion coupling variables for handling nonlocal couplings 178 Eytelwein's equation 522 F fanout 475 feeder clamp deformation of 503 Fermi level 443 ferrohydrodynamics 342 fifth-order element Argyris 192 FitzHugh-Nagumo equations 101

fluid valve 208

fluid-structure interaction 399, 408

fractal dimension 271 free surface flow 247 friction 522

- G gas dynamics 230 gate variable 102 Gauss' law 55 geophysics 258 grid settings 362 groundwater flow 258
- H heat capacity 331
 heat equation 277
 heat source 331
 Heat transfer

 application mode 330
 heat transfer 358
 Heat Transfer Module 122
 heating with a moving laser 278
 Helmholtz equation 20, 560
 higher-order elements 192, 350
 hydraulic head 270
 hydraulic jumps 143
 hydrogen atom 422
- ideality factor L in diode equation 450 Incompressible Navier-Stokes application mode 322 Incompressible Navier-Stokes application mode 358 initial value 447 integral constraints handling using ODE interface 178 integrating on boundary 372 integration order Argyris element 195 interface conditions 399 interpolation

bilinear 271 linear 271 inviscid flow 230 isospectral drums 568 isospectral membranes 568 iterative solver 63 IV characteristics 448

- K Karman vortex street 198
 KdV equation 128
 Korteweg 128
- L Lagrange elements 350 Lagrange multiplier 447 Lagrange multipliers 249 laminar flow analytical expression for 33 Landau-Ginzburg equations 101 laser beams 278 lift coefficient 204 linear interpolation 271 loading model 373 Lorentz force 302 Lorentz' equation 302
- M magnet brake 302 magnetic field intensity 53 magnetic flux density 53 magnetic susceptibility 349 magnetite 156 Marangoni convection 377 materials library 90 MATLAB using functions in 273 Maxwell's equations 442, 476 merging components 299 micromixer 219 microrobot 389 microrobot model 389 mixer, laminar static 219

modal analysis 496 model pulley 522 Model Library I Model Library models 5 model simplification 3D to 2D 357 symmetry planes 358 momentum balance 358, 377 MOS transistor 474 moving mesh 247 Moving Mesh application mode 247, 414 MPa units 496 muffler acoustics model of 10 multicomponent diffusivities 44 multidisciplinary model 329 multiphysics extended 310 multiphysics modeling 358 multiphysics models Marangoni convection 377 microrobot 389 vibrations in milk containers 399 N nanocarrier systems 342 NASTRAN mesh 496 Navier's equation 399

Navier-Stokes equations 192, 238
 nojac operator 183
 nonlinear solver 447
 nonlocal couplings
 handling using extrusion coupling variables 178
 ODE interface

for handling integral constraints 178 ODEs 238 on-off controller 331 ordinary differential equation 303, 414

orthogonal eigensolutions 572 output times 374 P p2-p1 elements 201 parametric solver 447 peristaltic pump 408 permeability 74, 303 Petrov-Galerkin streamline diffusion 112. 232 physics mode Structural mechanics 507 PID controller 317 Planck's constant 422, 432 Plane Stress pulley model 522 plasma dynamics 171 Poisson's ratio 400 polytropic gas 230 potential between cylinders 5 potential flow 270 preconditioner algebraic multigrid 63 pressure 19, 270, 400 principal quantum number 434 probability density function 424 for electron's position 422 process control 317 pulley 522

- Q quadratic element 309 quadrupole lens 83 quantum dots 432 quantum mechanics models 421 quantum numbers 423
- R radiation view factor 119
 rate constant
 in transport and adsorption model 32
 regular triangular mesh 118

remanent flux density 349 resonator chambers 10 Reynold's equation 522 Reynolds equation 270 RF Module 53 Richardson extrapolation 570, 574 rock fracture 270 rock fracture flow 270 room, eigenfrequencies of 19

s Saint-Venant's shallow water equations 142 sand grain, falling in water 237 Schrödinger equation 432 for electron energy levels 422 seepage velocity 260 semiconductor diode model of 442 semiconductor models MOS transistor 474 set difference 365 shallow water equations 142 shock tube 230, 420 Shockley-Read-Hall recombination 443, 476 simplified model 3D to 2D 357 symmetry planes 358 Simulink 304, 310 exporting to 336 skin effect 74 sloshing tank 247 smooth step function 414 solitons 128 solute transport 258 sparsity-pattern plot 178 speed of sound 19 Spherical symmetry 156 spherical symmetry 156

SPICE parameters 459 Stefan's constant 119 stream function 192 streamline diffusion 112, 118, 231 stress analysis of pulley 522 structural mechanics application mode 507 Structural Mechanics Module 495 subsurface fluid flow 258 surface concentration in transport and adsorption model 31 surface reactions 30 surface tension 377 symmetry in loads and geometry 504 symmetry boundary condition 443, 476 symmetry planes 358 T tangential derivative variable 34, 150 telegraph equation 135 test functions 249 thermal conductivity 330 thermal controller 329 thermal expansion coefficient 378 thermostat 329 thin structures modeling of 150 time-dependent simulation 373 time-harmonic solution 19 time-stepping output times from 374 townsend unit 181 transistor bipolar 457 transmission loss, in muffler 10 transport of species 30 transport problem 111 tubular reactor 30

two-term Boltzmann equation 171 typographical conventions 3

U ultraweak contributions 118 ultraweak term 114 unit systems MPa 496 upwinding stabilization 113

 variable tangential derivative 34
 vibrations in milk containers 399
 viscous transfer 378

W water waves

modeling of 128 wave equation 19 weak constraint for computation of viscous forces 199 weak constraints 249, 447 weak form 112 Weak Form, Point application mode 313 weak mode for points 313 weak term 116 wetting-layer system 432 Winslow smoothing 248

Y Young's modulus 400